Karlsruhe Institute of Technology

Faculty of Informatics
Institute for Anthropomatics
Chair Prof. Dr.-Ing. R. Stiefelhagen
Facial Image Processing and Analysis Group

Multi-View Facial Expression Classification

DIPLOMA THESIS OF

Nikolas Hesse

ADVISORS

Dr.-Ing. Hazım Kemal Ekenel
Dipl.-Inform. Hua Gao
Dipl.-Inform. Tobias Gehrig

MARCH 2011

Facial Image Processing and Analysis Group
Institute for Anthropomatics, Prof. Dr.-Ing. Rainer Stiefelhagen
Karlsruhe Institute of Technology
Title: Multi-View Facial Expression Classification
Author: Nikolas Hesse

Nikolas Hesse
Ludwig-Wilhelm-Str. 16
76131 Karlsruhe
nhesse@t-online.de

Statement of authorship

I hereby declare that this thesis is my own original work which I created without illegitimate help by others, that I have not used any other sources or resources than the ones indicated and that due acknowledgement is given where reference is made to the work of others

Karlsruhe, 30. March 2011

. . . .
(Nikolas Hesse)

Abstract

In the last decade, facial expression recognition has attracted more and more interest of researchers in the computer vision community. Facial expressions are a form of nonverbal communication, used to exchange social and emotional information in human-human-interaction. By detecting the expression of a human and reacting proactively, many applications could benefit from automatic facial expression recognition systems, e.g. human-computer-interfaces or security systems. Further applications for expression recognition lie in driver safety and social sciences. In order to use facial expression recognition systems in real-world situations, it is essential to recognize expressions not only from frontal face images, but also from images containing faces with pose variations. Furthermore, facial landmarks have to be located automatically in order to use local appearance features for representing the face.
In this work, a multi-view facial expression recognition system based on Active Appearance Models (AAMs) is established, which automatically finds facial landmark points by fitting a pose-dependent AAM to the input face image. Different features are extracted from the AAM, and appearance descriptors are computed at the located facial feature points using Scale-Invariant Feature Transform (SIFT) and Discrete Cosine Transform (DCT). On these features, feature selection is performed using the F-score feature selection method. The resulting feature vectors are then used for training pose-dependent linear multi-class Support Vector Machine (SVM) classifiers, which recognize six different expression classes for 13 different poses (frontal + six left + six right). Extensive experiments are performed on the BU-3DFE database to evaluate the expression recognition system. Recognition rates are measured for single feature types as well as for combinations of features. Results show, that a combination of DCT features and normalized landmark coordinates, extracted from the fitted AAM, achieves the highest recognition accuracy. Recognition rates for appearance features extracted at automatically located landmarks are compared to those of appearance features extracted at groundtruth landmarks. The influence of the intensity level of expressions on recognition accuracy is also explored.

Kurzzusammenfassung

In den letzten Jahren ist das Interesse von Forschungsgruppen im Bereich des Maschinensehens an automatischen Systemen zur Erkennung von Gesichtsausdrücken stetig gestiegen. Gesichtsausdrücke sind eine Form der nonverbalen Kommunikation, mit deren Hilfe soziale und emotionale Informationen zwischen Menschen ausgetauscht werden. Viele Anwendungen können von automatischen Gesichtsausdruckserkennungssystemen profitieren, indem sie auf die erkannten Gesichtsausdrücke proaktiv reagieren, z.B. Mensch-Maschine-Schnittstellen oder Überwachungssysteme. Weitere Anwendungsmöglichkeiten liegen in Fahrsicherheitssystemen oder den Sozialwissenschaften. Wenn solche Erkennungssysteme in realen Situationen verwendet werden sollen, ist es notwendig die Ausdrücke nicht nur auf Gesichtsbildern aus frontaler Ansicht zu erkennen, sondern auch auf Bildern aus anderen Blickwinkeln. Außerdem müssen die charakteristischen Gesichtspunkte automatisch gefunden werden, wenn lokale Bildmerkmale zur Repräsentation des Gesichtes benutzt werden sollen.

In dieser Arbeit wird ein System zur Erkennung von Gesichtsausdrücken aus verschiedenen Blickwinkeln entwickelt. Durch das Anpassen eines posenabhängigen Active Appearance Models (AAMs) an ein gegebenes Gesichtsbild werden charakteristische Gesichtspunkte automatisch lokalisiert. Aus dem AAM werden dann unterschiedliche Merkmale extrahiert und Bildmerkmale werden an den gefundenen Gesichtspunkten berechnet. Dazu werden Verfahren der skaleninvarianten Merkmalstransformation (SIFT) und der Diskreten Kosinustransformation (DCT) verwendet. Dann wird auf die Merkmalsvektoren das F-score Merkmalsauswahlverfahren angewandt. Mit den resultierenden Vektoren werden posenabhängige lineare multiklassen Support Vector Machines (SVM) trainiert, welche sechs Gesichtsausdrücke aus insgesamt 13 verschiedenen Blickwinkeln erkennen.

Umfassende Experimente werden auf der BU-3DFE Datenbank zur Evaluation des Gesichtsausdruckserkennungssystems durchgeführt. Die Erkennungsraten werden sowohl für einzelne Merkmalstypen, als auch für Kombinationen von Merkmalstypen gemessen. Ergebnisse zeigen, dass eine Kombination von DCT-Merkmalen und normalisierten Gesichtspunktkoordinaten, welche aus dem eingepassten AAM extrahiert wurden, die höchste Erkennungsrate erzielt. Die Erkennungsraten von Bildmerkmalen, welche an automatisch lokalisierten Gesichtspunkten berechnet wurden, werden mit denen von Merkmalen, welche an von Hand markierten Gesichtspunkten extrahiert wurden, verglichen. Zudem wird der Einfluss der Intensität der Gesichtsausdrücke auf die Klassifizierungsergebnisse untersucht.

Contents

1 Introduction **1**
- 1.1 Motivation . . . 2
- 1.2 Goals . . . 2
- 1.3 Thesis overview . . . 2

2 Related work **5**
- 2.1 Facial expression recognition based on geometry of facial features . . . 5
- 2.2 Appearance-based expression recognition . . . 7
- 2.3 Model-based facial expression recognition . . . 9
- 2.4 Multi-view facial expression recognition . . . 12

3 Theoretical Background **17**
- 3.1 Active Appearance Models . . . 17
  - 3.1.1 Model formulation . . . 17
  - 3.1.2 AAM Fitting . . . 19
  - 3.1.3 Pose-dependent AAMs . . . 26
- 3.2 Appearance Features . . . 26
  - 3.2.1 Scale-Invariant Feature Transform . . . 26
  - 3.2.2 Discrete cosine transform . . . 28
- 3.3 Feature Selection - F-score . . . 29
- 3.4 Support Vector Machines . . . 30
  - 3.4.1 General SVMs . . . 30
  - 3.4.2 Multi-class SVMs . . . 32

4 Methodology **33**
- 4.1 Training step . . . 33
- 4.2 Automatic facial expression analysis - framework . . . 36
  - 4.2.1 Face acquisition . . . 36
  - 4.2.2 Feature extraction and representation . . . 37
  - 4.2.3 Facial expression recognition . . . 39

5 Evaluation **41**
- 5.1 Data . . . 41
- 5.2 Experiment Setup . . . 42

5.3 Results 43
    5.3.1 AAM features 43
    5.3.2 SIFT/DCT features 45
    5.3.3 Combination of features 49
    5.3.4 Influence of intensity on recognition 52

6 Conclusion 55

Bibliography 57

List of Figures

3.1 lllustration of linear shape variation 18
3.2 Illustration of linear appearance variation 19
3.3 Example for AAM model instantiation 20
3.4 Illustration of the forwards compositional algorithm 22
3.5 Illustration of the inverse compositional algorithm 24
3.6 Illustration of a SIFT descriptor 27
3.7 Illustration of DCT basis functions 28
3.8 Illustration of zig zag scanning 29
3.9 Example of a Support Vector Machine 31
3.10 Example of a multi-class SVM 32

4.1 Overview of the proposed system 34
4.2 Illustration of AAM training 35
4.3 Illustration of SVM training process 36
4.4 Example of patches around facial landmark points for SIFT features . 37
4.5 Example of patches around facial landmark points for DCT 38

5.1 Example of BU-3DFE data 42
5.2 Example of BU-3DFE data including landmark points 42
5.3 Example subject from the BU-3DFE database showing different levels of intensity 42
5.4 Illustration of recognition accuracies across different poses for AAM features 45
5.5 Illustration of recognition accuracies across different poses for appearance features 47
5.6 Illustration of recognition accuracies across different poses for appearance features extracted at groundtruth landmarks 49
5.7 Illustration of recognition accuracies across different poses for combinations of features 51
5.8 Illustration of accuracies for different poses at different intensity levels 53

List of Tables

5.1 Recognition accuracies for different expressions for AAM features . . 44
5.2 Recognition accuracies for different poses for AAM features 44
5.3 Recognition accuracies for different expressions for appearance features 46
5.4 Recognition accuracies for different poses for appearance features . . 46
5.5 Recognition accuracies for different expressions for appearance features extracted at groundtruth landmarks 48
5.6 Recognition accuracies for different poses for appearance featurs extracted at groundtruth landmarks 48
5.7 Recognition accuracies for different expressions for combinations of features 50
5.8 Recognition accuracies for different poses for combinations of features 50
5.9 Complete confusion matrix for combination of shape coordinates with DCT 51
5.10 Recognition rates for classification of data containing expressions at different levels of intensity 52

List of Algorithms

1 Inverse compositional algorithm 24
2 Simultaneous inverse compositional algorithm 25
3 F-score feature selection algorithm 30

List of Abbreviations

AAM	Active Appearance Model
ASM	Active Shape Model
AU	Action Unit
BBN	Bayesian Belief Net
BU-3DFE	Binghamton University 3D Facial Expression Database
DCT	Discrete Cosine Transform
FACS	Facial Action Coding System
FSEL	Feature Selection
GDA	Generalized Discriminant Analysis
GND	Groundtruth
HoG	Histogram of oriented Gradients
LBP	Local Binary Patterns
LDA	Linear Discriminant Analysis
LPP	Locality Preserving Projection
LPQ	Local Phase Quantisation
MU	Motion Unit
NB	Naive Bayes
NN	Nearest Neighbor
PCA	Principal Component Analysis
RCM	Region Covariance Matrix
SFAM	Statistical Facial Feature Model
ShC	Shape Coordinates

SIFT	Scale-Invariant Feature Transform
SVM	Support Vector Machine
TOP	Three Orthogonal Planes
VTK	The Visualization Toolkit

1. Introduction

Over the last years, automatic facial expression recognition systems have become a more and more important field of research in the computer vision community. A big part of human communication is done through facial expressions. By interpreting the expression of the face, one can tell the emotional state of another person much faster than by using words. Consequently, automatic facial expression recognition systems can lead to big improvements in human-centered human-machine interaction, so that in the future, robots/machines can understand human behaviour and react proactively. Further applications lie in security, driver safety and social sciences as a tool to analyse human affective behavior.

Facial expression recognition uses visual information to divide facial motion and facial feature deformations into different abstract classes. Misleadingly, the term *emotion recognition* is often used as a synonym for expression recognition. Opposed to expression recognition, emotion recognition is an attempt to *interpret* facial expressions, but emotions are influenced by many factors, and are not necessarily displayed through facial expressions. Ekman and Friesen [1] introduce 6 *basic emotions*, i.e. anger, disgust, fear, happiness, sadness and surprise, "that possess each a distinctive content together with a unique facial expression". These basic emotions are often used in facial expression recognition systems as expression classes and are usually called *prototypic* or *basic expressions*. When creating a database using these 6 classes, the problem appears, that persons performing the different expressions have different understandings of how a certain expression should look, which leads to different results for the same classes. Therefore, the data has to be labelled by human experts afterwards to ensure the consistency of classes.

Opposed to the approach of recognizing complete expressions, there is a widely used system available called *FACS* (facial action coding system [2]) for detecting so called *action units* (AUs). This approach is solely based on appearance and does not try to *interpret* facial deformations. Activity of facial muscles or muscle groups and their intensities are measured and assigned to different predetermined AU classes. These classes can be divided into upper and lower face action units, with upper face AUs concentrating on eyes, their surrounding areas and eyebrows, while lower face AUs focus on deformations of mouth and cheeks.

Automatic facial expression analysis is a challenging task, as there are many factors that can make the same expression look very different. Changes in age, ethnicity, gender and facial hair will have a big influence on classification results, but even when the shown person is the same, occlusions or pose and lighting variations can cause big problems.

1.1 Motivation

Most existing facial expression recognition systems only work well on near-frontal faces. For using an automatic facial expression recognition system in real-world environments, the capability of dealing with non-frontal poses is essential. With the creation of a new 3D facial expression database [3], the number of multi-view facial expression recognition systems has grown. Many of these systems utilize hand-labelled points for feature extraction [4, 5, 6, 7, 8, 9, 10], which is not an option for application in real-world situations. Another drawback is the fact that in most systems only 5 poses are considered for expression recognition. Hence, in this thesis a system is developed which automatically extracts facial feature points from a given input image and recognizes facial expressions for 13 poses.

1.2 Goals

The goal of this thesis is to establish an automatic multi-view facial expression recognition system that works well in real world situations. Therefore, it has to be able to process images containing faces at arbitrary view angles up to profile view. The face must be found and the locations of the facial feature points have to be extracted automatically. A numerical representation of an expressive face has to be established, in order to distinguish between the different expression classes. Also, classifiers have to be trained to output the recognized expression in the end.

Since the appearance and shape of a face vary if the view angle changes, it is not sufficient to model faces from different poses by one model only. Hence, pose-dependent models will be used for describing shape and appearance of a face for each pose seperately. Consequentially, one expression classifier is built for each pose to recognize six basic expressions, namely anger, disgust, fear, happiness, sadness and surprise.

1.3 Thesis overview

In this work, a multi-view expression recognition system is built, which receives a 2D image containing a face as input. After performing pose estimation and face detection, a pre-trained pose-dependent *Active Appearance Model* (AAM) [11] is fitted to the input face. Several features are extracted from the AAM: shape and appearance parameters, and facial landmark points on which appearance descriptors, namely *Scale-Invariant Feature Transform* (SIFT) [12] and *Discrete Cosine Transform* (DCT) [13], are computed. Afterwards, feature selection is performed on the feature vector containing the extracted features in order to reduce the dimensionality and to improve the learnability. This feature vector is then fed to the previously trained pose-dependent multi-class *Support Vector Machine* (SVM) [14], which finally outputs the recognized expression class.

For creating the system described above, pose-dependent AAMs and pose-dependent SVMs have to be trained. Pose-dependent means, that for each pose one AAM/SVM is trained. After the pose of the input face is determined, the corresponding AAM, respectively SVM, is selected for further process. In this thesis, the view angle varies between -90 and +90 degrees for horizontal head-rotation with steps of 15 degrees, which results in 13 AAMs and 13 SVMs.

For validating the proposed system, experiments are conducted on the BU-3DFE database [3]. Recognition results are produced for different kinds of features, e.g. shape features extracted from the fitted AAM or appearance features computed at facial landmark points. It is investigated if combining shape and appearance features can improve recognition accuracy, and classification results for appearance features extracted at automatically selected landmarks are compared to results based on groundtruth landmarks.

The remainder of this thesis is organized as follows. First, an overview of existing facial expression recognition systems is given in section 2. In section 3, the theoretical backgrounds for the methods and algorithms used in this thesis are presented. Then, the facial expression recognition system developed in this thesis is introduced in section 4, which afterwards is evaluated through extensive experiments (section 5). Finally, in section 6, conclusions are drawn.

2. Related work

A wide variety of approaches to facial expression recognition exists. In this section, an overview of several state-of-the-art systems is given, which are divided into following categories:

- Facial expression recognition based on geometry of facial features
- Appearance-based facial expression recognition
- Model-based facial expression recognition
- Multi-view facial expression recognition.

2.1 Facial expression recognition based on geometry of facial features

When using geometric features for expression recognition, classification is based on locations of facial landmark points and distances between them. Therefore, landmark coordinates have to be labelled by hand or extracted automatically from the input image.

Tang and Huang present a facial expression recognition approach based on properties of line segments in [15]. Features for expression classification are normalized distances and slopes of facial features, which are computed from 3D facial landmark points. From each face, 96 features are extracted and used for recognizing six expression classes, namely anger, disgust, fear, happiness, sadness and surprise. Experiments on the BU-3DFE database using groundtruth landmark points show an average recognition accuracy of 87,1% using multi-class SVM classifiers.

In a similar approach in [16], the same authors use distances between facial landmark points for classification. This time, two types of feature selection are performed on these features. First, human experts select 24 features, which they consider relevant, by hand. The second method is an automatic feature selection, maximizing “the average relative entropy of marginalized class-conditional feature distributions”, which means, that distances between all possible pairs of landmark points are considered, and those with the biggest discrimination power are selected. On the automatically selected feature vector, Principal Component Analysis (PCA) is performed. Then, vectors obtained from both methods are fed to an AdaBoost classifier with different weak classifiers (Nearest Neighbor (NN), Naive Bayes (NB) and Linear Discriminant Analysis (LDA)). For experiments, data of 60 subjects is taken from the BU-3DFE database, half of which are female and the other half are male. For each of these 60 subjects, the corresponding neutral face features are subtracted from the expression face features as a preprocessing step. 54 subjects are used for training the classifiers and 6 for testing. Using the manually selected features, recognition rates are as following: 93.6% (NN), 93.8% (NB), and 91.8% (LDA), while for automatically selected features recognition accuracies of 94.8% (NN), 90.8% (NB), and 95.1% (LDA) are obtained.

A lot of research in this direction has been done by Soyel and Demirel. In [17], they propose using 3D facial feature distances for expression recognition. By making use of the symmetry of the face, they find the optimal number of facial feature 3D points to be eleven, from which six characteristic distances are computed. These are: eye opening distance, eyebrow height, mouth opening, mouth height, lip stretching and normalization (distance between outermost points on left/right face contour). These distances serve as input for a multilayer-perceptron-based neural network classifier. Experiments are conducted on the BU-3DFE database, from which 60 subjects are taken, showing seven expressions (basic expressions + neutral). For evaluation, this dataset is arbitrarily divided into training set (54 subjects) and test set (6 subjects) ten times and for each fold, classification is done. Results of this system show an average recognition accuracy of 91,3%.

The same authors continue work on this approach in [18], changing some parameters of the system. Hence, 23 facial landmark 3D points are used to compute six distance features as above, with ’height of mouth’ replacing ’width of mouth’ and ’openness of jaw’ replacing ’normalization’. Again, the BU-3DFE is used for experiments, using the same setup as above. This time, a probabilistic neural network is used for classification, which shows an average recognition rate of 87,8%.

A slightly different approach is presented in [19]. Here, normalized distances between all pairs of 83 facial landmark points, given in the BU-3DFE database, form a feature vector. From this vector, the most discriminating features are selected using Non-dominated Sorted Genetic Algorithm II for feature selection. The output from feature selection is fed to a probabilistic neural network. Experiments use data of 420 3D models from the BU-3DFE database, of which half are male, and the other half are female, containing seven expression classes. For evaluation, data is split into training set (336 models) and test set (84 models). Classification shows an average recognition rate of 88,18%.

In [20], Soyel and Demirel present an improved feature selection technique for the system presented above. On the feature vector containing all distances between the 83 landmark points, PCA is performed for dimensionality reduction. From the resulting principal components, an optimal subset according to the Fisher-criterion is searched. Into this subset, LDA is computed, leaving a (c-1)-dimensional discriminant subspace, with c being the number of classes. The experimental setup from [19] is used here as well. A probabilistic neural network produces an average recognition rate of 88,5%, using the presented feature selection method. Additionally, a decision-tree based probabilistic neural network classifier, using a coarse-to-fine scheme is proposed, which divides the seven expression classes into three groups containing following expressions: group 1: surprise; group 2: anger, sadness and neutral; group 3: disgust, fear and happiness. Therefore, in the coarse step, a new sample is assigned to one of the groups, whereas in the fine step, the final classification is performed. This approach shows superior results, with an average recognition accuracy of 93,7%.

2.2 Appearance-based expression recognition

Appearance-based approaches extract information about facial expressions from a given image, without having extensive knowledge about the object of interest. Algorithms using this method are typically fast and simple, running filters and classifiers on an image [21].

Zhu et al. introduce dynamic cascades with bidirectional bootstrapping for selecting positive and negative samples from video sequences for action unit detection in [22]. In order to classify action units, facial landmark points have to be found, which is accomplished by using AAMs for face alignment, from which 66 facial feature points are extracted. In addition to the points from the AAM, points from some other areas of interest are used, e.g. nasolabial furrows. In order to obtain accurate positions of these additional points over a sequence, a backward piecewise affine warp is applied. In the next step, appearance features are extracted. Before representing the appearance features by computing SIFT descriptors on the landmark points, the input face is normalized by registering it with respect to an average face, using similarity transform. Additionally, difference of scale, in-plane-rotation and transformation among the images are eliminated. Afterwards, positive and negative sample sets are repeatedly learned and updated by utilizing bidirectional bootstrapping in combination with the training of dynamic cascade detectors.

Based on [22], a system using segment-based SVMs to detect action units in video sequences is developed in [23]. This system combines the two main approaches in this area of research, which are static modeling and temporal modeling. Static modeling examines each video frame independently and represents a discriminative classification problem, while temporal modeling arranges frames into sequences and is typically represented by a variant of dynamic Bayesian networks. Segment-based SVMs consider "AU detection as a problem of detecting temporal events in a time series of visual features". Advantages of this approach are the modeling of dependencies between features and length of AUs, the possibility of utilizing all segments for training and the fact that no assumptions about the structure of the AUs are needed.

In [24], the use of an unsupervised technique for clustering facial events is explored. The goal is to automatically find facial actions in video sequences to accumulate them in clusters. Therefore, facial features are detected and tracked by using Active Appearance Models over a sequence of images. For normalization, each face is registered with respect to an average face. Then, shape and appearance features are generated for upper and lower face separately from data obtained from the AAM. Shape features include distance between inner brow and eye, distance between outer brow and eye, height of eye, height of tip, height of teeth and angle of mouth corners. For appearance features, SIFT descriptors are extracted from eleven points around the outer outline of the mouth and from five points on the eyebrows. On these features, PCA is utilized to perform feature selection. The resulting feature vector represents the expressive input face and is further processed by Aligned Cluster Analysis in order to cluster different facial expressions.

A real-time face detection and facial expression recognition system is presented in [25]. Haar-filters (Viola & Jones) are utilized to locate faces, which then are rescaled and transformed into a Gabor magnitude representation, using a set of Gabor filters at eight orientations and five spatial frequencies. These high dimensional feature vectors are used for training classifiers. For classification of seven expressions (basic expressions + neutral), SVM, Adaboost and a combination of both, called AdaSVM, are compared. First, for each expression, one SVM is trained to discriminate between the current and all other expressions. Then, by selecting the classifier with the maximum margin for the test example, the class decision is made. For evaluation, leave-one-subject-out cross-validation is performed. Further experiments are conducted, using linear, polynomial and RBF kernels with Laplacian and Gaussian basis functions, showing best results for linear SVMs and Gaussian RBF kernel SVMs. These results are then compared to Adaboost, which selects a subset of individual Gabor filters as features. After optimizing threshold and scale parameters of each filter, the feature performing best on the boosted distribution is chosen. Leave-one-group-out cross-validation is used for evaluation, because Adaboost is remarkably slower than SVMs. Finally, the two classification methods above are combined, resulting in a classifier called AdaSVMs. Adaboost is used to select Gabor features, which then serve as input for training SVMs. Experimental results show a recognition improvement of 3,8% over Adaboost and 2,7% over SVMs. Further investigation reveals, that by doubling the resolution and increasing the number of Gabor wavelets from five to nine, classification accuracy is increased even more.

Zhang et al. compare geometry-based features and Gabor wavelet-based features for expression classification in [26]. Geometry-based features used are coordinates of 34 facial landmark points, which are labelled manually on a given face. At these points, multi-scale and multi-orientation Gabor wavelet coefficients are extracted. Experiments using a two-layer perceptron for dimensionality reduction and classification show that Gabor wavelet-based features achieve much better results than geometric features. Combining both types of features shows no significant improvement of recognition accuracy.

Yang et al. [27] seek to avoid using AUs and rather use compositional features around AU areas to interpret facial expressions. A given input face image is split into local patches according to the locations of the AUs, and from each patch, local appearance is represented by haar-like features. From a combination of some of these features, compositional features are built, which are then processed by a boosting learning procedure to construct a classifier.

In [28], Jiang et al. explore the use of Local Binary Pattern descriptors for AU detection. For single images, Local Binary Patterns (LBP) and Local Phase Quantisation (LPQ) are used, while for video sequences Local Binary Patterns from Three Orthogonal Planes (LBP-TOP) and an extension to LPQ, which is called Local Phase Quantisation from Three Orthogonal Planes (LPQ-TOP), are compared. In the proposed system, first, the face is located by an adapted Viola and Jones face detector. Then, head rotations and scale variations are removed by registering the static image, respectively the first frames of the sequence. The frames of a sequence are then aligned automatically and the sequence/static image is divided into small blocks, from which LBP-TOP and LPQ-TOP, respectively LBP and LPQ features are extracted. Obtained LBP histograms are concatenated to represent the image or sequence, before feature selection is performed. Experiments using SVMs as classifiers to detect nine upper face AUs show that LPQs generally achieve higher recognition rates than LBPs, and that LPQ-TOP outperforms all other tested approaches.

Another approach utilizing LBPs for expression recognition is presented in [29], where Shan and Gritti propose to learn LBP histogram bins in order to discriminate different facial expressions. The bins offer a compact and discriminative representation of expressive faces, but since not all bins contribute to recognition performance, the relevant ones are selected by Adaboost. Multi-scale LBP histogram bins are also evaluated, showing an improvement over single scale LBPs.

2.3 Model-based facial expression recognition

Opposed to the appearance-based approaches, model-based methods describe the face by modeling its shape and its appearance. The downside of this approach is, that for the construction of the models, it is necessary to label the facial landmarks by hand. Therefore, having a big amount of training data implies a lot of manual labor.

In [30], Sebe et al. create a database which contains videos showing authentic facial expressions, and introduce a real-time facial expression recognition system. A face is tracked by a piecewise Bézier volume deformation tracker, which constructs a 3D wireframe model of the face. Landmark points are selected interactively in the first frame of a sequence, and a face model consisting of 16 surface patches is warped to fit the selected landmarks. From the fitted model, head motion and facial deformations are extracted. By applying template matching to successive frames, 2D image motions are measured, from which 3D motions are estimated. A magnitude representation is used, which connects each feature motion to a simple deformation of the face, resulting in a set of so called motion units (MUs). MUs are similar to action units and represent the activation of a facial region, as well as the direction and intensity of the motion. These MUs are used as input for classification. Experiments are conducted on the generated database and on the Cohn-Kanade database, applying a wide variety of classifiers. Best results are obtained by k-Nearest-Neighbor-classifiers.

Ramanathan et al. [31] introduce a system which uses 3D morphable models to recognize facial expressions. Morphable Expression Models are built from 3D face meshes which are generated by a 3D face scanner. The morphing parameters represent facial deformations with respect to the neutral face and are therefore used as features for expression classification. Each set of morphing parameters for a given expressive face defines a point in the expression space, where different clusters are formed for expressions neutral, happiness, sadness and anger. A new sample is classified by mapping it to the expression space through morphing and then deciding which cluster it belongs to.

Lucey et al. present a system for facial action recognition by employing AAM derived facial representations for classification in [32]. AAM features are obtained by AAM tracking a face and extracting 2D shape and appearance as well as 3D shape information from the AAM. Experimental results are generated, using different classifiers, which are k-Nearest-Neighbor and Support Vector Machines with RBF kernel.

Another approach using AAMs is described in [33], where an AAM is fitted to a new face, from which similarity normalized shape and canonical normalized appearance features are extracted. These features, as well as a combination of both, are used for training SVM classifiers. Experiments show, that the combination of both feature types outperforms the use of only one.

An approach to manifold based expression recognition using Active Shape Models (ASM) is presented in [34], which describes a new representation for tracking and recognition of facial expressions based on manifold embedding and probabilistic modeling in the embedded space. Therefore, video data is mapped to a low dimensional expression manifold in a feature space, described by facial landmarks. Then, a Gaussian mixture model is applied to cluster the data in the expression space. Each of the clusters is represented by an ASM on top of which a particle filter is used for tracking facial deformations and recognizing facial expressions, together in one probabilistic framework.

Tong et al. introduce a unified probabilistic facial action model based on a Dynamic Bayesian network in [35] to represent the spatio-temporal relationships between rigid and non-rigid facial motions in video sequences. The relationship information from the model is combined with visual information extracted from the given image. Experiments indicate that the proposed method shows superior performance in comparison to Adaboost and a semantic AU model.

In [36], Mpiperis et al. present a system for 3D facial expression recognition and expression-invariant 3D face recognition using bilinear models which represent the contribution of expression and identity components to the facial appearance. Therefore, through deformable 3D models, point correspondences among faces are determined in order to align facial feature locations. Then, an asymmetric bilinear model is fitted to the deformable model parameters by estimating expression control matrices and identity control vectors that minimize the total squared reconstruction error. These estimations are used to build a Maximum Likelihood classifier in order to estimate the likelihood of deformable model parameters for each expression. A new sample is then assigned to the expression class with the highest likelihood. Experiments for 3D expression recognition on the BU-3DFE database show an average recognition rate of 90.5%.

Lee and Elgammal propose an approach to facial expression analysis in video sequences using nonlinear shape and appearance models in [37]. Dynamics of facial expressions are modeled using low-dimensional manifold embedding. Non-linear generative models and kernel mapping are utilized for learning non-linear shape and appearance models in low-dimensional spaces which represent the facial deformations during facial expressions. The model parameters, which are iteratively estimated, are then used for classification.

A framework for 3D expression recognition in videos is built in [38], which represents expression sequences as path consisting of clusters on the expression manifold, starting from a point that represents the neutral expression. The generalized expression manifold is built by transferring the facial deformations from the video to a standard model. Representing the transition between clusters and paths, a probabilistic model is learned. Then, the probability for each facial expression is modeled as a mixture density with clusters as mixture centers. From a given test sequence, facial deformations are projected to the standard model and posterior probabilities are calculated for all expressions, assigning the expression class with the highest probability to the sequence.

Zhao et al. propose a framework for automatic 3D expression recognition based on a Bayesian Belief Net (BBN) and a Statistical Facial Feature Model (SFAM) in [39]. In order to automatically find facial landmark points, a previously learned SFAM is fitted to a new 3D face. From the SFAM, different features are extracted, which are landmark locations, face texture and face shape. To improve recognition results, additionally, shape index is calculated to model local surface curvature information, and multi-scale LBPs are used to improve the representation of face texture. Through the BBN, beliefs for different expressions are learned, and a new expressive face is classified according to its calculated belief. Experiments on the BU-3DFE database show an average recognition rate of 82,3% for automatically located facial landmarks and 87,2% for manually located landmarks.

In [40], an approach for modeling and tracking rigid and non-rigid 3D facial deformations from 2D video sequences is introduced. A face is represented by 42 facial landmark points, which are localized by a 2D active shape model in the video sequences. The landmarks are tracked over a sequence and then projected to 3D by using a previously learned face model. A 3D deformable face model is constructed from a combination of 1D nonlinear manifolds, where each manifold represents a mode of deformation or expression. The manifolds are learned offline from sequences of labelled expressions using Tensor Voting, which is used for estimating geometric information. Any expression is then represented by a linear combination of values along the manifold axes.

2.4 Multi-view facial expression recognition

Multi-view expression recognition systems extend frontal face expression recognition approaches in order to process expressive face images or video sequences at different view angles.

Hu et al. explore in [4] whether or not non-frontal view expressions can achieve higher recognition accuracy than expressions on frontal faces. Therefore, 2D images together with groundtruth landmark points are generated from the BU-3DFE database at five different views (0, 30, 45, 60, 90 degrees of head rotation). The geometric 2D displacement of the landmark points in expressive faces is calculated in comparison to the neutral face and normalized for each subject in the database. The locations of the normalized landmark points around eyes, eyebrows and mouth form a feature vector, which is used as input for classification. Different classifiers are evaluated, which are: Linear Bayes Normal Classifier, Quadratic Bayes Normal Classifier, Parzen classifier, SVMs with linear kernel and K-Nearest-Neighbor Classifier. The latter is used in combination with feature selection techniques PCA, LDA and Locality Preserving Projection (LPP). 5-fold-cross-validation is used for generating results, employing 80 subjects for training and 20 for testing. Experiments show, that highest recognition accuracy is achieved on non-frontal views between 30 and 60 degrees, with SVM classifiers showing best overall results with an average error rate of 33,5%. The authors conclude that the reason for non-frontal views to achieve better results than frontal view might be, that frontal faces contain redundancy due to the symmetry of the face, while faces rotated by around 45 degrees additionally contain depth information.

In [5], an approach to multi-view facial expression recognition is presented, which compares classification results for several appearance features, which are extracted at groundtruth facial landmark points. Techniques used for extracting features are: Histogram of Oriented Gradients (HoG), LBP and SIFT. Moreover, raw appearance intensity is used for comparison. Before doing classification with a Nearest-Neighbor-classifier, feature selection is performed, using LPP, PCA and LDA. Original data without feature selection is also classified for comparison. The system is divided into two stages. First, a view classifier determines the pose angle of the input image, with possible values being 0, 30, 45, 60 and 90 degrees. Then, for each pose, an expression classifier is trained to output one of the six basic expression classes. Experiments are conducted on the BU-3DFE database, which contains 3D models of 100 subjects, showing facial expressions at different levels of intensity. In the experiments, for each of these models, 2D images are taken from five different angles. Coordinates of the facial landmarks are extracted and saved as well. For evaluation, a 5-run two-fold cross-validation person-independent scheme is used, randomly splitting the 100 subjects of the database into two equally sized groups, using one group as training data and the other group as test data. First, the appearance features are extracted from the facial landmark points. Then, for dimensionality reduction, different feature selection methods are utilized. Finally, this data serves as input for the Nearest-Neighbor classifier. For each of the feature extraction methods, the results are given for the original data set as well as for data sets using LPP, PCA and LDA for feature selection. Error rates are displayed for raw appearance intensity, with the original data set achieving the best result with an average error rate of around 57%. Using HoG features, LPP data improves by around 22 percent compared to the original set and has the best result with an average error rate of around 32%. For LBP and SIFT, LPP outperforms the other reduction methods (and original set) by far, showing average error rates of around 35% (HoG) and around 27% (SIFT), while the other methods get error rates of around 54% on HoG features (except LDA: around 39%) and around 44% on SIFT features. Also, experiments based on a combination of SIFT+LPP, HoG+LPP and LBP+LPP classifiers are performed, showing results of 26,54% best average error rate. Another approach introduced in this paper is to build only one classifier with each possible combination of view and expression being considered a class. Since there are five views and six expressions, this adds up to 30 classes. The average error rate for HoG features increases by up to 10%. There are no results for SIFT and LBP features for this approach since this experiment exceeded the available computer capabilities.

In [6], a system for multi-view facial expression recognition under Bayes theoretical framework is developed. Features for classification are extracted by computing SIFT descriptors at 83 hand-labelled landmark points on face images. Each descriptor is a 128-dimensional feature vector. Hence, for representing a complete face, a 10624-dimensional feature vector is received, which is then reduced to 500 dimensions using PCA. Multi-class expression recognition is performed by “minimizing an estimated closed-form Bayes error”. For experimental validation, data from the BU-3DFE database at five different views is used. Results are produced by running ten trials, each trial randomly splitting 100 subjects into two sets, the training set containing 80 subjects, the test set 20 subjects. By averaging results of all runs, an error rate of 21,65% is received. For comparison, k-Nearest-Neighbor classifiers are trained using SIFT features, processed by LDA respectively PCA, resulting in an average error rate of 23,1% respectively 33,84%.

Zheng et al. introduce a novel system for emotion recognition from arbitrary view facial images in [41], using a region covariance matrix (RCM) representation of face images. First, the face region is detected on a given input image. Then, this region is split into patches and at the center of each patch, a SIFT feature vector is extracted. By computing the covariance of the SIFT vectors, a region covariance matrix is received. This approach has the advantage, that neither face alignment nor facial feature localization is necessary. The authors also present a new discriminant analysis theory for selecting the most relevant features, which carry the most discriminative information from the facial feature vectors by minimizing an estimated multi-class Bayes error, derived under the Gaussian mixture model. Also, an algorithm to solve the optimal discriminant vectors is proposed. Evaluation is conducted on the BU-3DFE database, from which 2D images are extracted. Only expression images showing the highest level of intensity are used in this study. Face poses are varied using yaw angles (-45, -30, -15, 0, 15, 30, 45 degrees) and pitch angles (-30, -15, 0, 15, 30 degrees). Therefore, 100 (subjects) $\times$ 6 (expressions) $\times$ 7 (yaw angles) $\times$ 5 (pitch angles) = 21000 images are generated. The dataset is split into five equally-sized sets and five-fold cross-validation is performed, using four sets for training and one set for testing. Lowest error rate is achieved for frontal face without any pose variation (25%). Lowest average error rates are shown for yaw = 0 degrees with pitch varying (28,27%), and for pitch = 0 degrees with yaw varying (28,1%).

In [42], a system using LBPs for multi-view facial expression recognition is presented. In this approach, images are divided into 64 sub-blocks, and similarities between the blocks are compared. Thus, local texture as well as global shape of a face image is captured. Then, a histogram of LBP features is computed, forming a feature vector which is fed to a pose-dependent classifier. Experiments are conducted on two different datasets. First, images are extracted from the BU-3DFE database at 5 different poses (0, 30, 45, 60, 90 degrees) and 4 different resolutions (32×44, 44×62, 64×88 and 80×110). Pose estimation is performed to select a pose-dependent SVM for later classification. Different LBP approaches are used to compute LBP features, which are:
-LBP^{riu2}: Uniform rotation invariant local binary patterns
-LBP^{ri}: Rotation invariant local binary patterns
-LBP^{gm}: Uniform local binary patterns obtained from gradient magnitude image
-LBP^{u2}: Standard local uniform binary patterns with a neighborhood of 8 pixels and a radius of 1 pixel
-LBP^{ms}: Multi-scale local binary patterns where radius varies from 1 to 8 pixels

-*LGBP*: Local binary patterns extracted from gabor images, where 40 different gabor images are composed from applying gabor kernels at different scales and orientations.

Results show a maximum difference of less than 3% for varying resolutions. *LGBP* and LBP^{ms} perform best, with an average accuracy of 67,96%, respectively 65,02%. By combining these two, results improve further to 71,1%. Influence of intensity levels are also reviewed, indicating average accuracies of 56,83%, 68,83%, 73,04% and 77,67% from low to high intensities for *LGBP*. The second database, on which experiments are performed, is Multi-PIE. It contains images from 337 subjects, which are predominantly male (70%). The database contains a variety of ethnicities and ages. The 100 subjects selected display following facial expressions: neutral, smile, surprise, squint, disgust and scream. First, a Viola & Jones face detector is run on the input image. Then, the experiment proceeds similar to experiments on the BU-3DFE database. Results are shown for LBP^{ms} and *LGBP*, achieving 73,98%, respectively 80,17% accuracy. Best results appear at 15 degree pose.

A regression-based approach to multi-view facial expression recognition is presented in [7]. Since there is much more training data available for frontal faces than for non-frontal faces, Rudovic et al. propose to map facial landmark points from non-frontal view to frontal view using different state-of-the-art regression methods, which are: Linear Regression, Support Vector Regression, Relevance Vector Regression and Gaussian Process Regression. The system is organized as follows. Given a 2D image containing a non-frontal view face, first, the head pose is estimated and assigned to a pre-defined pose class. Then, 39 facial landmark points are localized, using state-of-the-art-methods. The goal is then to find mapping functions that transform the landmark points from arbitrary poses to frontal pose. After the functions are found, they are used to predict the locations of the landmark points in frontal view. The predicted points are then input to a frontal view expression classifier. Finally, the classifier outputs the recognized expression class. Experiments are conducted on CMU Multi-PIE database, using 4 different views, namely 0, 15, 30 and 45 degrees, where 4 expressions are to be recognized by linear SVMs.

Taheri et al. [8] present an approach leading towards view-invariant expression analysis using analytic shape manifolds, which avoids dependency of facial deformations on a camera coordinate frame. A face shape is considered an equivalence class across view changes, rather than a vector containing feature information. The rigid and non-rigid deformations of facial landmarks are decoupled, so that the deformations caused by head rotation can be ignored. Generally, a 2D image of a face shows a perspective projection from 3D to 2D points and therefore, the authors claim the projective shape-space to be well suited for modeling facial geometry as equivalence classes independent of head poses. Hence, invariance to camera angle changes can be achieved. For better understanding, projective shapes are approximated by using affine shapes. A sequence of faces showing an expression can be represented by a sequence of points in the Grassmann manifold. Facial deformations are then modeled by geodesics on the manifold, “where a geodesic is a path of shortest length on the manifold between two given points”. Geodesics are described by velocity vectors on the tangent plane at the starting point. Experiments are conducted on different datasets (CK, Bosphorus, talking face) to show that this approach achieves good recognition results in classifying AUs as well as basic emotions. Geodesics are learned for each AU and each emotion, on which then, LDA (and kernel LDA for Grassmann space) is performed, before classification is done using SVMs. Results of algorithms on Grassmann space and Euclidean space are compared, showing that classification on Grassmann space outperforms classification on Euclidean space.

A system for recognition of profile view action units is developed in [9], using video sequences which show profile faces displaying expressions. Each sequence is divided into onset, apex and offset. In the first frame, 15 facial feature points are initialized by hand, which are then tracked throughout the whole sequence using particle filtering. For each frame, the movement of the landmark points is measured with respect to the location of the points in the first frame. Therefore, two parameters are defined:
- $up/down(P)$, which measures upward and downward movements of point P,
- $inc/dec(PP')$, which measures increase and decrease of the distance between P and P'.

The movement of the facial feature points, represented by these parameters, is used for classifying 27 AUs.

The application of view-based 2D + 3D Active Appearance Models in combination with generalized discriminant analysis (GDA) for expression classification is proposed by Sung and Kim in [10]. In order to build 2D + 3D AAMs, facial feature points are hand-labelled in training images for three different poses: left, frontal and right. For each pose, 2D shape and appearance models are built, and from the 2D models for different views a 3D shape model is constructed. Hence, 3 pose-dependent 2D AAMs are obtained. For a given input image, the pose is estimated and the corresponding AAM is selected, which is then fitted to the image. From the fitted AAM, 2D appearance and 3D shape coefficients are extracted, which are then processed by a GDA method, which transforms the coefficients into feature vectors in the 2D appearance feature spaces and 3D shape feature space, respectively. The feature vectors are then concatenated and fed to another GDA, which transforms the concatenated vector into an integrated facial expression feature vector. A Malahanobis distance based classifier finally determines the facial expression class. Experiments on four different expressions, namely neutral, happiness, surprise and anger, show similar recognition rates for the three different poses. The frontal model seems to deal better with classifying expressions from another pose, compared to left and right pose models.

3. Theoretical Background

In this work, a multi-view expression recognition system is built, which receives a 2D image containing a face as input. After performing pose estimation and face recognition, a previously trained Active Appearance Model is fit to the input face. Several features are extracted from the AAM: shape parameters, appearance parameters and facial landmark points, on which then appearance descriptors (SIFT/DCT) are computed. Dimensionality reduction is performed on the extracted feature vectors through feature selection, and finally, these vectors are fed to a SVM classifier. In this section the theoretical backgrounds of these methods and algorithms are presented.

3.1 Active Appearance Models

Active Appearance Models were introduced in [11] by Cootes et al. and are used for matching a statistical model of object shape and appearance to a new image. AAMs are often utilized for modeling faces or other deformable objects. In this section, a general model formulation is given, presenting definitions of shape and appearance modeling [43]. Then, different algorithms for AAM fitting are introduced and explained, and finally, pose-dependent AAMs are specified.

3.1.1 Model formulation

Independent AAMs model shape and appearance separately, as described below. There exist also *combined* AAMs, which use a single set of parameters for describing shape and appearance. This formulation is more general and needs less parameters to model the same visual phenomenon in comparison to independent AAMs. However, there are also disadvantages of this approach. It can be no longer assumed that eigen-shapes and eigen-appearances are respectively orthonormal. Furthermore, since combined AAMs restrict the choice of fitting algorithms, the inverse compositional algorithm used in this work would not be applicable. Therefore, this section focuses on independent AAMs.

Shape

The shape $\mathbf{s}$ is defined by a mesh and the vertex locations of the mesh, i.e. the coordinates of the v vertices: $\mathbf{s} = (x_1, y_1, x_2, y_2, ..., x_v, y_v)^T$. AAMs allow linear shape variation, which means that the shape can be described by a base shape $\mathbf{s}_0$ plus a linear combination of n shape vectors $\mathbf{s}_i$:

$$\mathbf{s} = \mathbf{s}_0 + \sum_{i=1}^{n} p_i \mathbf{s}_i, \tag{3.1}$$

with p_i being shape parameters. See Figure 3.1 for illustration.

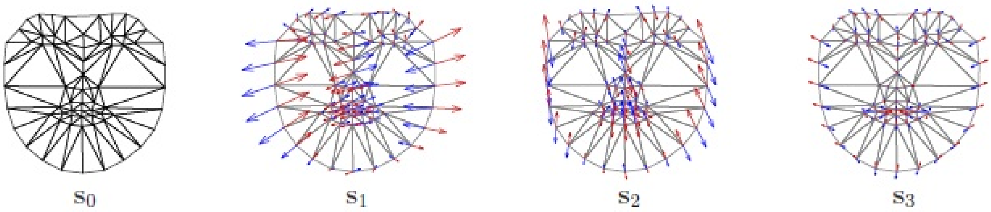

Figure 3.1: Illustration of linear shape variation. AAM shape is described by the base shape $\mathbf{s}_0$ plus a linear combination of shape vectors $\mathbf{s}_i$, here i = 1, 2, 3. Image taken from [43].

For building an AAM, training images containing hand-labelled landmark points are needed. In the standard approach, PCA is applied to the training meshes. The base shape $\mathbf{s}_0$ is the mean shape, and the vectors $\mathbf{s}_i$ are the n eigenvectors corresponding to the n largest eigenvalues.

Appearance

The appearance of an AAM is an image $A(\mathbf{x})$, defined over the pixels $\mathbf{x}$ lying within the base mesh $\mathbf{s}_0$. Similar to shape, AAMs allow linear appearance variation, which means that the appearance can be described by the base appearance $A_0(\mathbf{x})$ plus a linear combination of m appearance images $A_i(\mathbf{x})$:

$$A(\mathbf{x}) = A_0(\mathbf{x}) + \sum_{i=1}^{m} \lambda_i A_i(\mathbf{x}), \ \forall \mathbf{x} \in \mathbf{s}_0, \tag{3.2}$$

with λ_i being appearance parameters. An example is shown in Figure 3.2.

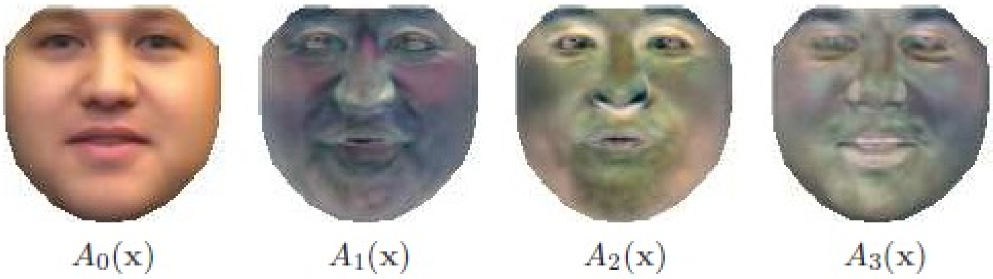

Figure 3.2: Illustration of linear appearance variation. AAM appearance is described by base appearance $A_0(x)$ plus a linear combination of appearance images $A_i(x)$, here i = 1, 2, 3. Image taken from [43].

By applying PCA to a set of shape normalized training images, A_0 and A_i are computed. The shape normalization is obtained by warping the training shape onto the mean shape $\mathbf{s}_0$. After triangulating the shape, a piecewise affine warp is defined between corresponding triangles in the training and base meshes. A_0 is the mean image, while the images A_i are the m eigenimages corresponding to the m largest eigenvalues.

Model instantiation

A requirement for instantiating a model is to have available both the shape $\mathbf{s}$, which is computed from the shape parameters $\mathbf{p} = (p_1, p_2, ..., p_n)^T$, and the appearance $A(\mathbf{x})$, which is computed from the appearance parameters $\lambda = (\lambda_1, \lambda_2, ..., \lambda_m)^T$. The AAM model instance is then generated by warping the appearance A from the base mesh $\mathbf{s}_0$ to the model shape $\mathbf{s}$. The pair of meshes $\mathbf{s}_0$ and $\mathbf{s}$ define the warp from $\mathbf{s}_0$ to $\mathbf{s}$ which is denoted as $\mathbf{W}(\mathbf{x};\mathbf{p})$. For each triangle in $\mathbf{s}_0$ a corresponding triangle in $\mathbf{s}$ exists. Between the members of any pair of triangles a unique affine warp is defined, which maps the points of one triangle to the other triangle. The complete warp is obtained by finding out for any pixel $x \in \mathbf{s}_0$ in which triangle it is located and then warping it with the affine warp for that triangle. This piecewise affine warp is denoted as $\mathbf{W}(\mathbf{x};\mathbf{p})$. The AAM model instance $M(\mathbf{W}(\mathbf{x};\mathbf{p}))$ is obtained by warping the appearance A from $\mathbf{s}_0$ to $\mathbf{s}$ using $\mathbf{W}(\mathbf{x};\mathbf{p})$:

$$M(\mathbf{W}(\mathbf{x};\mathbf{p})) = A(\mathbf{x}), \tag{3.3}$$

where M is a 2D image of the appropriate size and shape that contains the model instance. See Figure 3.3 for an example.

3.1.2 AAM Fitting

To fit an AAM to a given image, algorithms are developed with the objective to find the best matching AAM parameters in an efficient way. In [43], the concepts of AAM fitting together with an overview of several algorithms is presented.

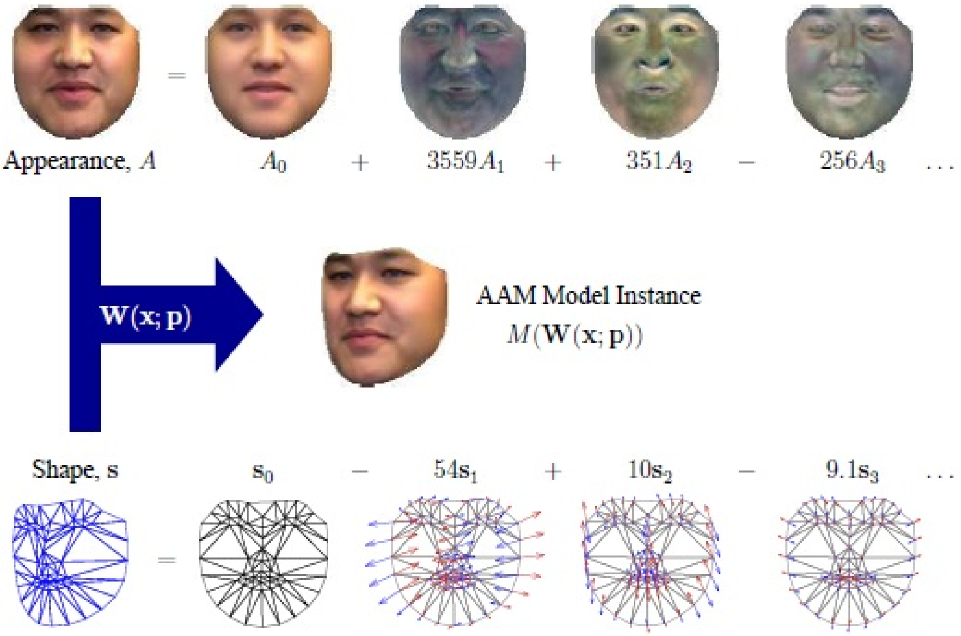

Figure 3.3: Example for AAM model instantiation. AAM model instance $M(\mathbf{W}(\mathbf{x};\mathbf{p}))$ is generated by warping the appearance A from the base shape $\mathbf{s}_0$ to the model shape s using the warp $\mathbf{W}(\mathbf{x};\mathbf{p})$. Image taken from [43].

Fitting goal

Given an input image $I(\mathbf{x})$, the goal of AAM fitting is to find optimal shape parameters p and appearance parameters λ of an AAM, so that the model instance $M(\mathbf{W}(\mathbf{x};\mathbf{p})) = A(\mathbf{x})$ is similar to $I(\mathbf{x})$. Therefore, the minimization of the error between model instance and input image is used as optimization criterion. For efficiency reasons, the error is computed in the base mesh $\mathbf{s}_0$ in the AAM coordinate frame instead of using the image coordinate frame. A pixel x lying in $\mathbf{s}_0$ has a corresponding pixel in the input image, which is defined by $\mathbf{W}(\mathbf{x};\mathbf{p})$. The appearance of the AAM at pixel $\mathbf{x}$ is defined by $A(\mathbf{x}) = A_0(\mathbf{x}) + \sum_{i=1}^{m} \lambda_i A_i(\mathbf{x})$ and the intensity of the input image at pixel $\mathbf{W}(\mathbf{x};\mathbf{p})$ is $I(\mathbf{W}(\mathbf{x};\mathbf{p}))$. The error to be minimized is the sum of squares of the differences between these two quantities over all pixels $\mathbf{x}$ in the base mesh $\mathbf{s}_0$:

$$\sum_{\mathbf{x}\in\mathbf{s}_0} \left[A_0(\mathbf{x}) + \sum_{i=1}^{m} \lambda_i A_i(\mathbf{x}) - I(\mathbf{W}(\mathbf{x};\mathbf{p})) \right]^2 . \tag{3.4}$$

Equation 3.4 is minimized with respect to $\mathbf{p}$ and λ simultaneously. This optimization is nonlinear in $\mathbf{p}$ but linear in λ. The error image is defined as:

$$E(\mathbf{x}) = \left[A_0(\mathbf{x}) + \sum_{i=1}^{m} \lambda_i A_i(\mathbf{x}) - I(\mathbf{W}(\mathbf{x};\mathbf{p})) \right] \tag{3.5}$$

and can be computed by first backwards warping the image I onto the base mesh $\mathbf{s}_0$ with $\mathbf{W}(\mathbf{x};\mathbf{p})$ and then subtracting it from the current appearance of the AAM.

Inefficient gradient descent algorithms

The expression in 3.4 can be minimized by applying standard gradient descent optimization algorithms, which use a principled, analytical algorithm and easily understandable convergence criterions. The drawback of this approach is that it is very slow. The slowness is caused by several complex computations, which have to be recalculated for every iteration, like computing partial derivates, Hessian and gradient direction.

Efficient ad-hoc fitting algorithms

In order to avoid these complex computations, more efficient algorithms make the simple assumption that there is a constant linear relationship between the error image $E(\mathbf{x})$ and additive increments to the shape and appearance parameters:

$$\Delta p_i = \sum_{\mathbf{x} \in \mathbf{s}_0} R_i(\mathbf{x}) E(\mathbf{x}) \quad \textit{and} \quad \Delta \lambda_i = \sum_{\mathbf{x} \in \mathbf{s}_0} S_i(\mathbf{x}) E(\mathbf{x}) \tag{3.6}$$

where $R_i(\mathbf{x})$ and $S_i(\mathbf{x})$ are constant images defined on the base mesh $\mathbf{s}_0$. Constant means, that $R_i(\mathbf{x})$ and $S_i(\mathbf{x})$ are independent from p_i and λ_i. Therefore, the computational cost is reduced. But since the assumption is incorrect, at the same time the fitting accuracy is decreased.

Matthews and Baker [43] proved, that there can not be an efficient gradient descent algorithm that solves for $\Delta\mathbf{p}$ and then updates the parameters $\mathbf{p} \leftarrow \mathbf{p} + \Delta\mathbf{p}$. A different way of updating the parameters is the *compositional* approach, in which the entire warp is updated by composing the current warp with the computed incremental warp with parameters $\Delta\mathbf{p}$, leading to the update rule:

$$\mathbf{W}(\mathbf{x};\mathbf{p}) \leftarrow \mathbf{W}(\mathbf{x};\mathbf{p}) \circ \mathbf{W}(\mathbf{x};\Delta\mathbf{p}). \tag{3.7}$$

Lucas-Kanade algorithm

The Lucas-Kanade algorithm is a forwards-additive algorithm, which searches the location of a constant template image in an input image by minimizing the sum of squared differences between the template $A_0(\mathbf{x})$ and the input image $I(\mathbf{x})$ with respect to $\mathbf{p}$:

$$\sum_{\mathbf{x}} [A_0(\mathbf{x}) - I(\mathbf{W}(\mathbf{x};\mathbf{p}))]^2, \tag{3.8}$$

where $\mathbf{W}(\mathbf{x};\mathbf{p})$ is a warp that maps the pixels $\mathbf{x}$ from the template to the input image and has parameters $\mathbf{p}$. Given an initial estimate of $\mathbf{p}$, the Lucas-Kanade algorithm solves iteratively for increments to the parameters $\mathbf{p}$ by minimizing following Equation with respect to $\mathbf{p}$, and then updates $\mathbf{p} \leftarrow \mathbf{p} + \Delta\mathbf{p}$:

$$\sum_{\mathbf{x}} [A_0(\mathbf{x}) - I(\mathbf{W}(\mathbf{x};\mathbf{p} + \Delta\mathbf{p}))]^2. \tag{3.9}$$

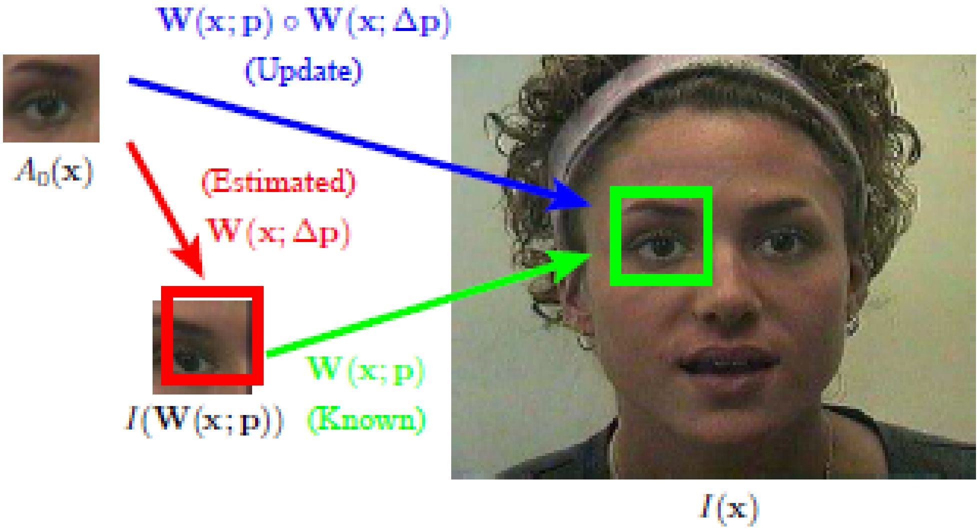
$\mathbf{W}(\mathbf{x};\mathbf{p}) \circ \mathbf{W}(\mathbf{x};\Delta\mathbf{p})$
(Update)
$A_0(\mathbf{x})$
(Estimated)
$\mathbf{W}(\mathbf{x};\Delta\mathbf{p})$
$\mathbf{W}(\mathbf{x};\mathbf{p})$
$I(\mathbf{W}(\mathbf{x};\mathbf{p}))$
(Known)
$I(\mathbf{x})$

Figure 3.4: Illustration of the forwards compositional algorithm. The template $A_0(\mathbf{x})$ is warped to the input image $I(\mathbf{x})$ with warp $\mathbf{W}(\mathbf{x};\mathbf{p})$. By minimizing the squared error between the warped template image and the input image, an incremental warp is found and composed with the original warp: $\mathbf{W}(\mathbf{x};\mathbf{p}) \leftarrow \mathbf{W}(\mathbf{x};\mathbf{p}) \circ \mathbf{W}(\mathbf{x};\Delta\mathbf{p})$. Image taken from [43].

Forwards compositional algorithm

Instead of updating the parameter p with estimated $\Delta\mathbf{p}$ offset, the compositional method computes an incremental warp $\mathbf{W}(\mathbf{x};\Delta\mathbf{p})$ to be composed with the current warp $\mathbf{W}(\mathbf{x};\mathbf{p})$. The minimization is over:

$$\sum_{\mathbf{x}} [A_0(\mathbf{x}) - I(\mathbf{W}(\mathbf{W}(\mathbf{x};\Delta\mathbf{p});\mathbf{p}))]^2 \tag{3.10}$$

and in the update step, the incremental and the current warp are composed:

$$\mathbf{W}(\mathbf{x};\mathbf{p}) \leftarrow \mathbf{W}(\mathbf{x};\mathbf{p}) \circ \mathbf{W}(\mathbf{x};\Delta\mathbf{p}). \tag{3.11}$$

Computing the solution for $\Delta\mathbf{p}$ in Equation 3.10 means that the incremental warp is computed in the 'image' direction. Composing it with the current warp (see Equation 3.11) results in the *forwards compositional* algorithm. An illustration is given in Figure 3.4.

Inverse compositional algorithm

The inverse compositional algorithm is a modification of the forwards compositional algorithm, where the roles of template image and sample input image are reversed. The incremental warp is computed with respect to the template $A_0(\mathbf{x})$, not with respect to $I(\mathbf{W}(\mathbf{x};\mathbf{p}))$. Therefore, the inverse compositional algorithm minimizes

$$\sum_{\mathbf{x}} [I(\mathbf{W}(\mathbf{x};\mathbf{p})) - A_0(\mathbf{W}(\mathbf{x};\Delta\mathbf{p}))]^2 \tag{3.12}$$

with respect to $\Delta\mathbf{p}$ and then updates the warp:

$$\mathbf{W}(\mathbf{x};\mathbf{p}) \leftarrow \mathbf{W}(\mathbf{x};\mathbf{p}) \circ \mathbf{W}(\mathbf{x};\Delta\mathbf{p})^{-1}. \tag{3.13}$$

Taking the Taylor expansion of equation 3.12 gives:

$$\sum_{\mathbf{x}} \left[I(\mathbf{W}(\mathbf{x};\mathbf{p})) - A_0(\mathbf{W}(\mathbf{x};\mathbf{0})) - \nabla A_0 \frac{\partial \mathbf{W}}{\partial \mathbf{p}} \Delta\mathbf{p} \right]^2. \tag{3.14}$$

Assuming that $\mathbf{W}(\mathbf{x};\mathbf{0})$ is the identity warp, the solution to this least squares problem is:

$$\Delta\mathbf{p} = \mathbf{H}^{-1} \sum_{\mathbf{x}} \left[\nabla A_0 \frac{\partial \mathbf{W}}{\partial \mathbf{p}} \right]^T [I(\mathbf{W}(\mathbf{x};\mathbf{p})) - A_0(\mathbf{x})], \tag{3.15}$$

where $\mathbf{H}$ is a Hessian matrix with I replaced by A_0:

$$\mathbf{H} = \sum_{\mathbf{x}} \left[\nabla A_0 \frac{\partial \mathbf{W}}{\partial \mathbf{p}} \right]^T \left[\nabla A_0 \frac{\partial \mathbf{W}}{\partial \mathbf{p}} \right]. \tag{3.16}$$

Since A_0 is constant and the Jacobian $\frac{\partial \mathbf{W}}{\partial \mathbf{p}}$ is always evaluated at $\mathbf{p} = \mathbf{0}$, most of the computations in Equations 3.15 and 3.16 have to be computed only once and can therefore be moved to a pre-computation step.

This results in a more efficient fitting algorithm, which is described step-by-step in Algorithm 1 and illustrated in Figure 3.5.

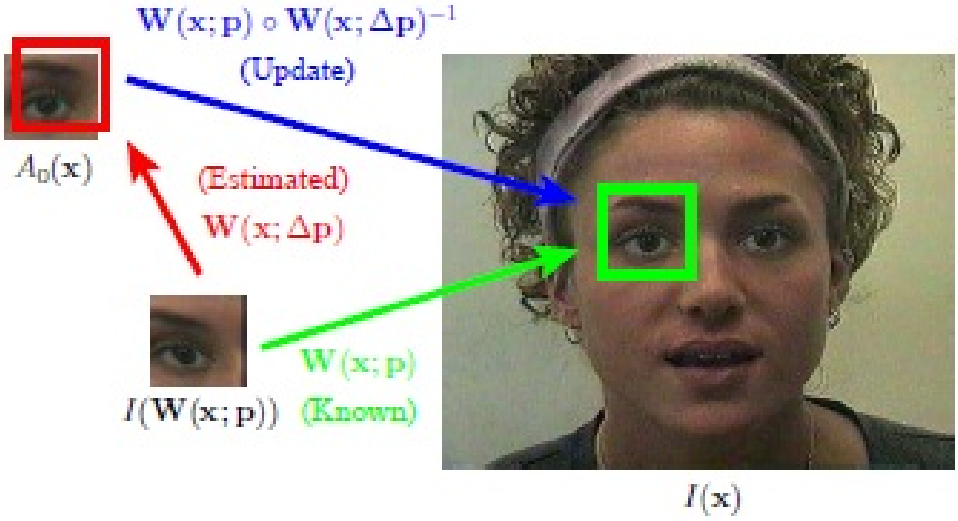
$\mathbf{W}(\mathbf{x};\mathbf{p}) \circ \mathbf{W}(\mathbf{x};\Delta\mathbf{p})^{-1}$
(Update)
$A_0(\mathbf{x})$
(Estimated)
$\mathbf{W}(\mathbf{x};\Delta\mathbf{p})$
$\mathbf{W}(\mathbf{x};\mathbf{p})$
(Known)
$I(\mathbf{W}(\mathbf{x};\mathbf{p}))$
$I(\mathbf{x})$

Figure 3.5: Illustration of the inverse compositional algorithm. Opposed to the forwards compositional algorithm, the incremental warp is computed with respect to the template $A_0(\mathbf{x})$, not with respect to $I(\mathbf{W}(\mathbf{x};\mathbf{p}))$. Therefore, the update step is composed by: $\mathbf{W}(\mathbf{x};\mathbf{p}) \leftarrow \mathbf{W}(\mathbf{x};\mathbf{p}) \circ \mathbf{W}(\mathbf{x};\Delta\mathbf{p})^{-1}$. Image taken from [43].

Algorithm 1 Inverse compositional algorithm

```
Pre-compute:
3: Evaluate the gradient $\nabla A_0$ of the template $A_0(\mathbf{x})$
4: Evaluate the Jacobian $\frac{\partial \mathbf{W}}{\partial \mathbf{p}}$ at $(\mathbf{x};\mathbf{0})$
5: Compute the steepest descent image $\nabla A_0 \frac{\partial \mathbf{W}}{\partial \mathbf{p}}$
6: Compute the Hessian matrix using Eq. 3.16
Iterate until converged:
1: Warp $I$ with $\mathbf{W}(\mathbf{x};\mathbf{p})$ to compute $I(\mathbf{W}(\mathbf{x};\mathbf{p}))$
2: Compute the error image $I(\mathbf{W}(\mathbf{x};\mathbf{p})) - A_0(\mathbf{x})$
7: Compute $\sum_{\mathbf{x}} \left[\nabla A_0 \frac{\partial \mathbf{W}}{\partial \mathbf{p}}\right]^T [I(\mathbf{W}(\mathbf{x};\mathbf{p})) - A_0(\mathbf{x})]$
8: Compute $\Delta\mathbf{p}$ using Eq. 3.15
9: Update the warp $\mathbf{W}(\mathbf{x};\mathbf{p}) \leftarrow \mathbf{W}(\mathbf{x};\mathbf{p}) \circ \mathbf{W}(\mathbf{x},\Delta\mathbf{p})^{-1}$
```

Simultaneous inverse compositional algorithm

At the beginning of this section, the goal of AAM fitting is set to minimize Equation 3.4 simultaneously with respect to shape parameters $\mathbf{p}$ and appearance parameters λ. Therefore, the simultaneous inverse compositional algorithm iteratively minimizes

Algorithm 2 Simultaneous inverse compositional algorithm

```
Pre-compute:
3: Evaluate the gradient $\nabla A_0$ and $\nabla A_i$ for $i = 1, ..., m$
4: Evaluate the Jacobian $\frac{\partial \mathbf{W}}{\partial \mathbf{p}}$ at $(\mathbf{x}; \mathbf{0})$
Iterate until converged:
1: Warp $I$ with $\mathbf{W}(\mathbf{x}; \mathbf{p})$ to compute $I(\mathbf{W}(\mathbf{x}; \mathbf{p}))$
2: Compute the error image $E_{app}(\mathbf{x})$ Eq. 3.22
5: Compute the steepest descent images $SD_{sim}(\mathbf{x})$ (Eq 3.18)
6: Compute the Hessian $\mathbf{H}_{sim}$ using Eq. 3.21 and invert it
7: Compute $\sum_{\mathbf{x}} SD^T_{sim}(\mathbf{x}) E_{app}(\mathbf{x})$
8: Compute $\Delta \mathbf{q} = -\mathbf{H}^{-1}_{sim} \sum_{\mathbf{x}} SD^T_{sim}(\mathbf{x}) E_{app}(\mathbf{x})$
9: Update the warp $\mathbf{W}(\mathbf{x}; \mathbf{p}) \leftarrow \mathbf{W}(\mathbf{x}; \mathbf{p}) \circ \mathbf{W}(\mathbf{x}, \Delta \mathbf{p})^{-1}$ and $\lambda \leftarrow \lambda + \Delta \lambda$
```

$$\sum_{\mathbf{x}} \left[A_0(\mathbf{W}(\mathbf{x}; \Delta \mathbf{p})) + \sum_{i=1}^{m} (\lambda_i + \Delta \lambda_i) A_i(\mathbf{W}(\mathbf{x}; \Delta \mathbf{p})) - I(\mathbf{W}(\mathbf{x}; \mathbf{p})) \right]^2 \tag{3.17}$$

simultaneously with respect to $\Delta \mathbf{p}$ and $\Delta \lambda = (\Delta \lambda_1, ..., \Delta \lambda_m)^T$ and then updates the warp $\mathbf{W}(\mathbf{x}; \mathbf{p}) \leftarrow \mathbf{W}(\mathbf{x}; \mathbf{p}) \circ \mathbf{W}(\mathbf{x}; \Delta \mathbf{p})^{-1}$ and the appearance parameters $\lambda \leftarrow \lambda + \Delta \lambda$.

Parameters $\mathbf{p}$ and λ are combined to form a $n + m$ dimensional column vector $\mathbf{q} = [\mathbf{p}, \lambda]^T$ and similarly, the update $\Delta \mathbf{q} = [\Delta \mathbf{p}, \Delta \lambda]^T$.

An $n + m$ dimensional steepest descent image SD_{sim} is then defined as:

$$SD_{sim}(\mathbf{x}) = (\nabla A \frac{\partial \mathbf{W}}{\partial p_1}, ..., \nabla A \frac{\partial \mathbf{W}}{\partial p_n}, A_1(\mathbf{x}), ..., A_m(\mathbf{x})), \tag{3.18}$$

where

$$\nabla A = \nabla A_0 + \sum_{i=1}^{m} \lambda_i \nabla A_i. \tag{3.19}$$

Then, the update is computed as

$$\Delta \mathbf{q} = -\mathbf{H}^{-1}_{sim} \sum_{\mathbf{x}} SD^T_{sim}(\mathbf{x}) E_{app}(\mathbf{x}), \tag{3.20}$$

where $\mathbf{H}^{-1}_{sim}$ is the inverse of the Hessian matrix

$$\mathbf{H}_{sim} = \sum_{\mathbf{x}} SD^T_{sim}(\mathbf{x}) SD_{sim}(\mathbf{x}) \tag{3.21}$$

and E_{app} is the error image between the warped input image and the model instance:

$$E_{app} = I(\mathbf{W}(\mathbf{x};\mathbf{p})) - \left[A_0(\mathbf{x}) + \sum_{i=1}^{m} \lambda_i A_i(\mathbf{x}) \right]. \tag{3.22}$$

The steepest descent images SD_{sim} depend on λ and therefore are not constant, which means, that they have to be recomputed each iteration. A summary of the algorithm is given in Algorithm 2.

3.1.3 Pose-dependent AAMs

In this thesis, facial expression recognition is performed for a variety of head poses. Using only one AAM for all different poses is not sufficient. Therefore, the poses are divided into different views, ranging from -90 to +90 degrees of horizontal head rotation with 15 degree steps. For each view, one AAM is trained, resulting in 13 AAMs. Hence, the pose of a given input image face has to be specified. A pose estimator processes the face in order to assign a pose class. According to this class, the corresponding AAM is selected for further application. In this work, the groundtruth pose is used for deciding which AAM to use, since pose estimation preferably should not have an influence on the recognition accuracy. Adding this component to the system is considered for future work. A pose estimator could fit different AAMs to a given face, compute the fitting error and then select the AAM with the smallest error. Another possibility is to locate a few landmark points on the input face image, e.g. eye centers, tip of the nose and mouth corners, and estimate the pose by finding correspondences between the landmark points and points on a previously learned face model.

3.2 Appearance Features

Two different approaches for representing facial expression images through appearance features are presented in this section, namely Scale-Invariant Feature Transform (SIFT) and Discrete Cosine Transform (DCT).

3.2.1 Scale-Invariant Feature Transform

SIFT, which was introduced by Lowe in [12], is an algorithm for extracting local appearance features from an input image. The general SIFT algorithm is often used for object recognition and has four processing steps:

- Scale-space extrema detection
- Keypoint localization
- Orientation assignment

- Keypoint descriptor extraction

Further information on the complete algorithm can be found in [44].

For applying SIFT to the facial expression recognition system in this work, not all steps are necessary, since facial landmark points are used as keypoints. Hence, the keypoint locations are extracted from the AAM, while scale and orientation of the keypoints are selected manually. The most important part of SIFT used here is the keypoint descriptors, which represent the local appearances of a given image.

To compute feature descriptors for given keypoints, several steps have to be performed. First, gradient magnitudes and orientations are sampled around each keypoint and the level of gaussian blur is selected according to the keypoint scale. Then, the coordinates of the descriptor and the gradient orientations are rotated relative to the orientation of the keypoint. To each sample point, a magnitude is assigned by a Gaussian weighting function, which prevents small changes in the window position leading to the occurrence of big changes in the descriptor. Also, gradients lying further away from the center of the descriptor are considered less significant. As shown in Fig. 3.6 on the left side, orientation histograms are created over 4 × 4 sample regions. For each histogram, eight directions are displayed, with the length of each arrow corresponding to the magnitude of the histogram entry. Each entry into a bin is multiplied by a weight of 1 - d for each dimension, where d is the distance of the sample from the central value of the bin. Finally, the descriptor is formed from a vector containing values of all orientation histogram entries corresponding to lengths of arrows. On the right side of Fig. 3.6, a 2 × 2 array of orientation histograms is shown. Further experiments indicate, that better results are received for 4 × 4 arrays of histograms with eight orientation bins each. In order to reduce the influence of illumination changes, the vector is normalized to unit length.

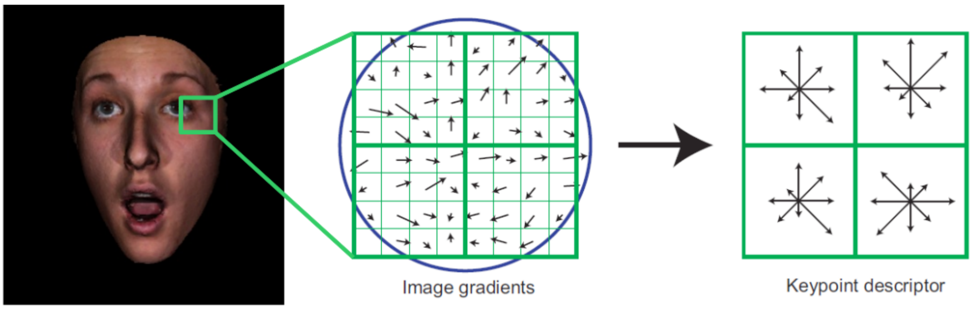

Figure 3.6: Illustration of a SIFT descriptor. From an area of an input image, image gradients are extracted, weighted by a Gaussian function, forming orientation histograms, from which the SIFT descriptor is constructed. Part of this image is taken from [44].

3.2.2 Discrete cosine transform

The discrete cosine transform (DCT) is a signal processing tool, which was introduced in [13]. Several variants of the DCT exist, but only the two-dimensional DCT used in this work will be described in this section. Further information on DCT can also be found in [45]. DCT can be used to represent local appearance features in a compact way, while preserving spatial relationships, outperforming methods like PCA and discrete wavelet transform when applied to face recognition [46]. The DCT for two-dimensional input $f(x, y)$ is defined as:

$$C(u,v) = \alpha(u)\alpha(v) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x,y) \cos\left[\frac{(2x+1)u\pi}{2N}\right] \cos\left[\frac{(2y+1)v\pi}{2N}\right] \tag{3.23}$$

for $u, v = 0, 1, ..., N-1$, where

$$\alpha(u) = \begin{cases} \sqrt{\frac{1}{N}} & \text{for } u = 0 \\ \sqrt{\frac{2}{N}} & \text{for } u = 1, 2, ..., N-1 \end{cases} .$$

An input image is divided into several blocks, and for each block, DCT coefficients are computed by applying DCT basis functions, which are shown in Fig. 3.7. The coefficient at the upper left (0,0) represents the average intensity value of the image, (1,0) the horizontal, (0,1) the vertical and (1,1) the diagonal changes in the image. The coefficients are ordered in a zig-zag pattern, which is displayed in Fig 3.8.

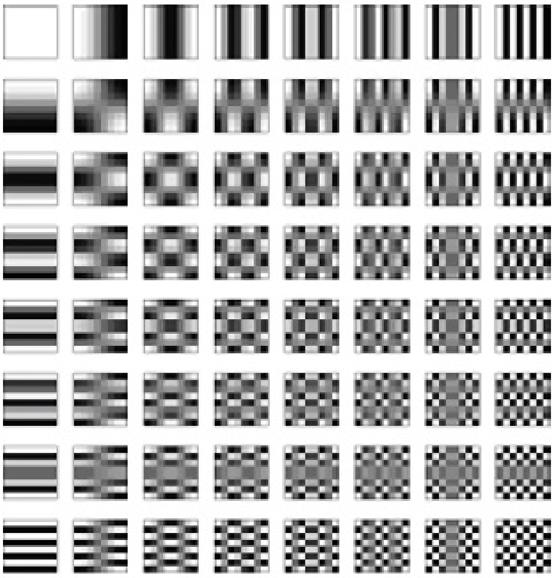

Figure 3.7: Illustration of DCT basis functions. When applied to an input image, the origin (0,0) at top left represents the average intensity value of the image, (1,0) the horizontal, (0,1) the vertical and (1,1) the diagonal changes in the image. Image taken from [47].

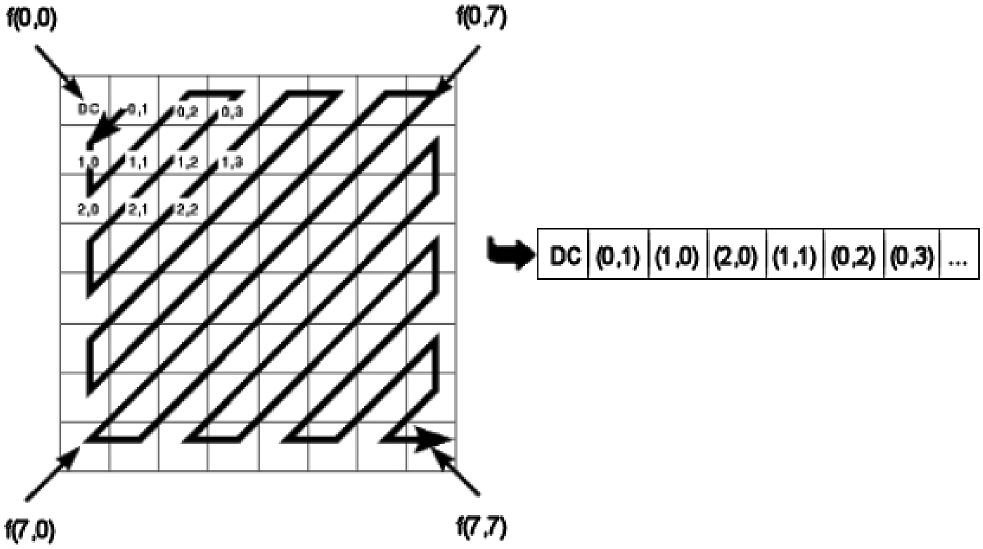

Figure 3.8: Illustration of zig zag scanning. The coefficients extracted from DCT are ordered in a zig zag pattern from top left to bottom right. Image taken from [47].

From each block, a subset of the ordered coefficients is selected, leaving out the first coefficient (at (0,0)), because it only indicates the average intensity value of the block. The selected coefficients are concatenated to form a feature vector for further processing of the image. In this work, DCT features are computed on facial landmark points on images of size 300 × 300 pixels, using 68 points for frontal poses and 36 points for side poses. Around each landmark point, a block of size 64 × 64 pixels is divided into four blocks of size 32 × 32 pixels, extracting the first 20 coefficients (except the first one) from each block.

3.3 Feature Selection - F-score

Sometimes thousands of features are used as input for a classifier. Some of them may be redundant or noisy. Therefore, feature selection picks the features relevant for classification, and drops the irrelevant ones. This results in a reduction of the data dimensionality, which increases the learnability as well as the computational efficiency.

In this work, the feature selection tool of LibSVM library called FSelect [48] has proven to produce best results. It uses the F-score technique to find the best subset of features.

F-score searches the subset of features that discriminates two sets of real numbers the best. Given training vectors $\mathbf{x}_k$, where $k = 1, ..., m$, and numbers of positive and negative instances n_+ and n_-, respectively, the F-score of the ith feature is defined as:

$$F(i) \equiv \frac{\left(\overline{\mathbf{x}}_i^{(+)} - \overline{\mathbf{x}}_i\right)^2 + \left(\overline{\mathbf{x}}_i^{(-)} - \overline{\mathbf{x}}_i\right)^2}{\frac{1}{n_+ -1}\sum_{k=1}^{n_+}\left(x_{k,i}^{(+)} - \overline{\mathbf{x}}_i^{(+)}\right)^2 + \frac{1}{n_- -1}\sum_{k=1}^{n_-}\left(x_{k,i}^{(-)} - \overline{\mathbf{x}}_i^{(-)}\right)^2} \tag{3.24}$$

where $\overline{\mathbf{x}}_i$ is the average of the ith feature of the complete data set and $\overline{\mathbf{x}}_i^{(+)}$ and $\overline{\mathbf{x}}_i^{(-)}$ are the averages of the ith feature of the positive and negative data sets, respectively. $x_{k,i}^{(+)}$ is the ith feature of the kth positive instance, and $x_{k,i}^{(-)}$ is the ith feature of the kth negative instance. In Eq. 3.24, the discrimination between the positive and negative sets is represented by the numerator, while the denominator describes the discrimination within each of the two sets. Features achieving higher F-scores are likely to be more discriminative than features with low F-score. Hence, the F-score is used as feature selection criterion. The complete algorithm for F-score feature selection is presented in Algorithm 3.

Algorithm 3 F-score feature selection algorithm

```
1: Calculate F-score of every feature.
2: Use different possible thresholds to cut low and high F-scores.
3: for each threshold do
4:     Drop features with F-score below threshold.
5:     Randomly split the training data into $X_{train}$ and $X_{valid}$.
6:     Let $X_{train}$ be the new training data. Use the SVM procedure to obtain a
       predictor; use the predictor to predict $X_{valid}$.
7:     Repeat the steps above five times, and then calculate the average validation
       error.
8: end for
9: Choose the threshold with the lowest average validation error.
10: Drop features with F-score below the selected threshold. Then apply the SVM
    procedure.
```

3.4 Support Vector Machines

The proposed expression recognition system is supposed to output an expression class in the end. That is, why a classification method is needed, which decides for given feature values, which class a new instance belongs to. In this system, Support Vector Machines, introduced in [14], are used, which are supervised learning methods that analyze data and recognize patterns.

3.4.1 General SVMs

General binary linear SVMs process labelled training feature vectors in a high-dimensional feature space, where vectors of two different classes are separated linearly by a hyperplane. An optimal hyperplane is found, if the distance between the hyperplane and the so called *Support Vectors*, which are the vectors defining the margin of the hyperplane. The hyperplane is determined by the Support Vectors, hence the name Support Vector Machine. An illustrating example is given in Figure 3.9.

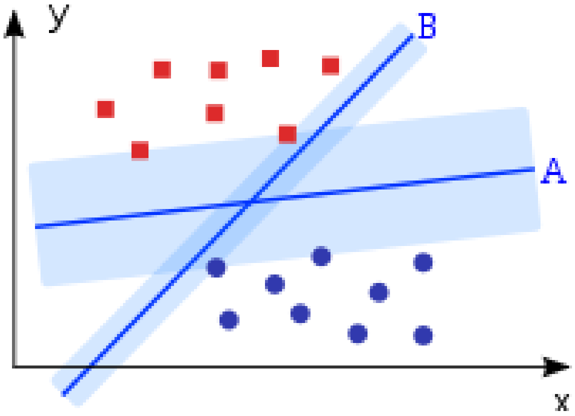

Figure 3.9: Example of a Support Vector Machine separating red squares and blue points. A: hyperplane optimally separating 2 classes, B: non-optimal hyperplane. Points/Squares lying closest to optimal hyperplane are called Support Vectors. Image taken from [49].

Any hyperplane can be written as a set of points x, satisfying

$$\mathbf{w} \cdot \mathbf{x} - b = 0, \tag{3.25}$$

where $\cdot$ is the dot product, $\mathbf{w}$ a normal vector to the hyperplane and $\frac{b}{\|\mathbf{w}\|}$ describes the offset of the hyperplane from the origin along $\mathbf{w}$. Hyperplanes spanned by the Support Vectors have the maximum distance to the optimal hyperplane and are defined by

$$\mathbf{w} \cdot \mathbf{x} - b = 1 \quad \textit{and} \quad \mathbf{w} \cdot \mathbf{x} - b = -1. \tag{3.26}$$

The distance between these two hyperplanes is $\frac{2}{\|\mathbf{w}\|}$. Consequently, to maximize the distance between these hyperplanes, $\mathbf{w}$ has to be minimized. After an optimal hyperplane in the feature space has been found, an unlabelled test sample is classified according to the side of the hyperplane it is located on.

There might be circumstances, where it is not possible to separate two classes linearly without error. If this is the case, it is possible to have a soft margin, which allows a minimal number of errors to occur. Another way of solving this problem is to apply the so called *kernel-trick*: the non-linear data is mapped to a high-dimensional space and the dot product is replaced by a non-linear kernel function. The mapping of the data to the high-dimensional space, where the data is linearly separable, can then be computed directly by using an appropriate kernel function. Therefore, a hyperplane in a high-dimensional space, which separates the data linearly, can be calculated implicitly.

3.4.2 Multi-class SVMs

In this work, six expression classes are to be recognized, which is why a multi-class classifier is needed. Multi-class SVMs handle this problem by combining several binary SVMs, using either one-versus-one or one-versus-all as training strategy. In this work, one-versus-one strategy is utilized, where for training purposes one class is considered positive and one other class negative. To get a classification result, a voting strategy is used, where for all pairs of classes the current feature vector is assigned to one of the two classes and finally, the class that receives most votes is considered the correct class. An illustration of one-versus-one multi-class SVMs is displayed in Figure 3.10.

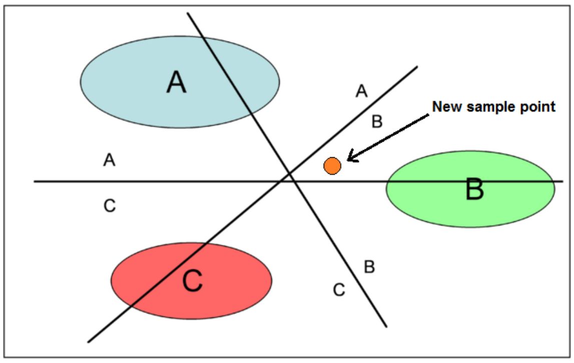

Figure 3.10: Example of a multi-class SVM. For each pair of classes, a separating hyperplane is learned. To assign a new sample to a class, it is classified by all pairs of classes, the votes/wins are counted and the sample is assigned to the class with most votes/wins. The one-versus-one classification in this example happens as follows: the first pair of classes is (A, B), the new sample lies on the 'B-side' of the separating hyperplane and therefore B gets one vote. The second pair of classes is (A, C) and the new sample is classified as A. The last pair is (B, C) classifying the sample as B. Summing up, A has one vote, B has two votes, C has zero votes; therefore, the new sample is classified as B. Image taken from [50].

4. Methodology

The goal of this thesis is to establish a multi-view facial expression recognition system based on Active Appearance Models. The proposed system is illustrated in Fig. 4.1, which shows the different processing steps. In this section, an overview of the system is given, followed by a detailed description of each of the systems components.

4.1 Training step

Before the recognition of an unknown face image can be performed, several preliminary steps have to be taken.

- Data generation
  Since there is no expression database available that contains 2D facial expression images with a wide variety of head poses, a database containing 3D models of expressive faces is used in this work. Therefore, 2D images have to be extracted from the 3D models by rotating them horizontally and extracting 2D images and 2D landmark coordinates. This is done for angles from -90 to 90 degrees, with 15 degree steps.

- Data division
  Data is divided into three equally sized sets: *AAM training*, *SVM training* and *testing*, without overlapping subjects.

- AAM training
  First, pose-dependent person-independent AAMs are trained: for each pose, corresponding 2D images with landmark points across all expressions are used for training the model. The training process is illustrated in 4.2. Pose is determined by groundtruth label.

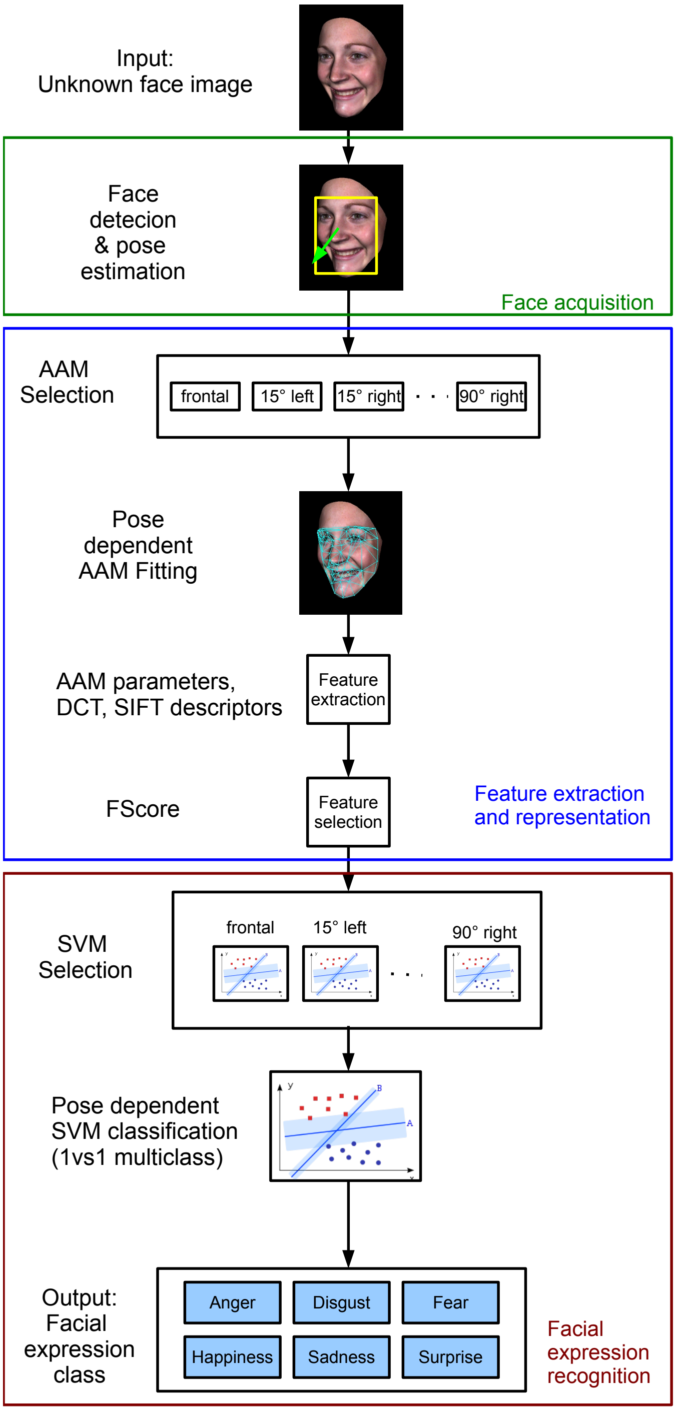

Figure 4.1: Overview of the proposed system. Input is an unknown face image, on which processing steps face acquisition, feature extraction and representation, and facial expression recognition are performed, outputting the recognized expression class.

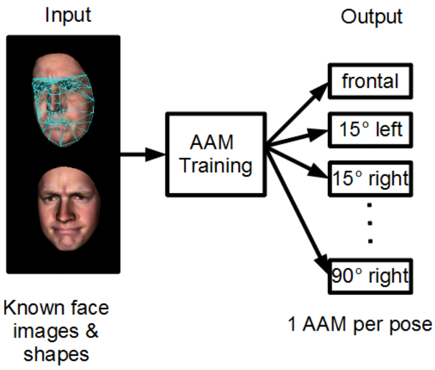

Figure 4.2: Illustration of AAM training. Face images with landmark points are used for training one AAM per pose.

- Face detection
  Face detection is needed for the initialization of AAM fitting. In order to avoid influences of face detection on expression recognition results, the groundtruth scale and location from the input face are used in this work.

- Fitting SVM training data
  AAM fitting is performed on every image in the SVM training set using a pose-dependent AAM which is determined by groundtruth pose.

- Feature extraction
  From the fitted AAM, shape and appearance information, as well as landmark locations are received, on which appearance features are computed, resulting in high-dimensional feature vectors.

- Feature selection
  Dimensionality reduction is performed by running F-score feature selection. Feature vectors of fitted SVM training data are used as input, together with class labels. A list of relevant features is output and for each sample, a reduced set of feature values according to the list is given.

- Scaling
  All feature values are scaled to values between 0 to 1 and the ranges are saved for later scaling of test data.

- SVM Training
  For each pose, the relevant and scaled features from the SVM training set, together with corresponding class labels for each sample are used as input for the SVM training process, which is performed by LibSVM. Therefore, for each pose, one linear multi-class SVM is trained, with which later, new test samples are classified. The complete training process is displayed in Figure 4.3.

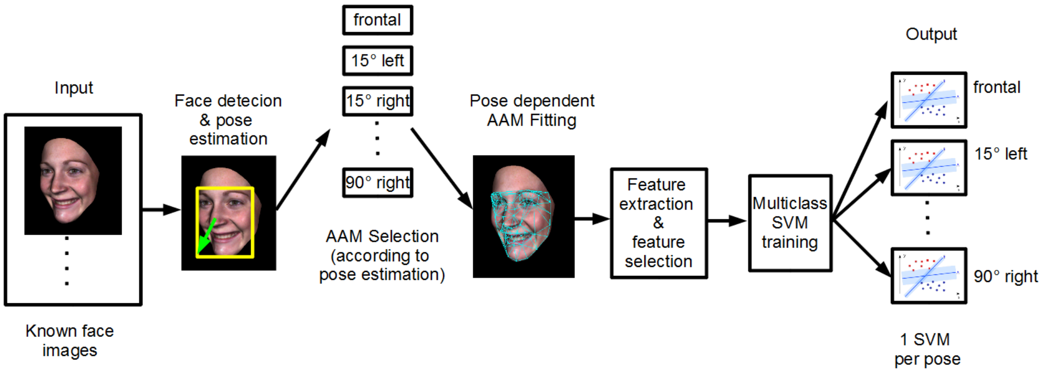
Input
Known face images
Face detecion & pose estimation
frontal
15° left
15° right
90° right
AAM Selection (according to pose estimation)
Pose dependent AAM Fitting
Feature extraction & feature selection
Multiclass SVM training
Output
frontal
15° left
90° right
1 SVM per pose

Figure 4.3: Illustration of SVM training process. AAM fitting is performed on images from SVM training set and features are extracted and used for training one SVM classifier per pose.

4.2 Automatic facial expression analysis - framework

In [21] and [51], a general framework for the automatic analysis of facial expressions is presented, on which the system architecture in this work is based. This framework is divided into the following tasks:

- face acquisition
- feature extraction and representation
- facial expression recognition

Given below is a detailed description of the different steps of the framework, including the methods and tools used for the execution of each task. An illustration of the proposed system is displayed in Figure 4.1.

4.2.1 Face acquisition

If any of the above problems are existent in given images, a normalization of the faces might be required before analyzing them.

Before features can be extracted from a face, the face has to be found in the given input image. By running a face detector on the image, locations and scales of the found faces are specified, usually displayed by a rectangle. Additionally, for multi-view expression recognition, it is also essential to know the degree of head rotation. This is accomplished by using a pose estimator, which outputs the detected pose in 15-degree-steps.

According to the detected pose an AAM is selected. For fitting the AAM to the new image, a starting position has to be given, which specifies location and scale of the AAM. This position is obtained from the face detector, and the AAM is initialized.

In this work, groundtruth information is used instead of face detection and pose estimation, since it is preferred to have no influence of these methods on recognition accuracy. Including face detection and pose estimation methods will be necessary for using this system in real world applications.

4.2.2 Feature extraction and representation

Having initialized the AAM at the determined starting position, fitting is performed using the simultaneous inverse compositional algorithm as described in section 3.1.2. After the fitting has converged, several features are extracted from the fitted AAM, which are: facial landmark coordinates, shape parameters and appearance parameters.

The 83 extracted facial landmark points include points on the face outline, and points around eyes, eyebrows, nose and mouth. For normalization, the shape is aligned using similarity transform, before 2D coordinates of landmark points are saved, resulting in a 166-dimensional feature vector for each face. The number of shape parameters varies for different poses, lying at around 55, while the number of appearance parameters is around 250.

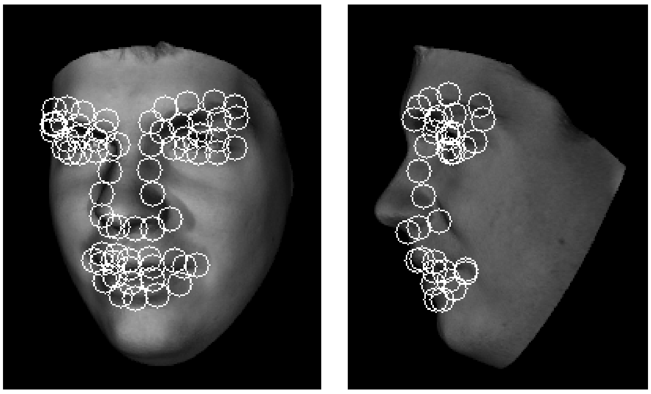

Figure 4.4: Example showing circles around facial landmark points for frontal poses (left image) and side poses (right image). Circles represent SIFT keypoints.

Keypoints for extracting appearance features are specified by selecting a subset of the extracted landmark points, containing only points at locations which are considered relevant for facial expressions. Therefore, points on the face outline are not used. Different subsets are used for different poses. For frontal poses (0-30 degrees) all landmark points except those on the face outline are used, resulting in 68 points. For side poses (45-90 degrees) only points lying on the visible side of the face are used, overall 36 points. An example displaying the selected points for frontal and side poses is given in Figure 4.4.

SIFT descriptors (see section 3.2.1) are computed at circluar regions with a diameter of 15 pixels around the obtained keypoints. An illustration is given in Figure 4.4. At each keypoint, 128 descriptors are extracted, resulting in a 8704-dimensional feature vector for frontal poses and a 4608-dimensional vector for side poses.

A second method applied for extracting appearance features is DCT (see section 3.2.2), which uses the same keypoints as SIFT. An area of size 64 × 64 pixels around each keypoint (see Figure 4.5) is split into 4 blocks of 32 × 32 pixels, and for each block the first 20 coefficients are saved, resulting in 80 coefficients per keypoint. Concatenating these coefficients produces 5440-dimensional (frontal), respectively 2880-dimensional (side) feature vectors.

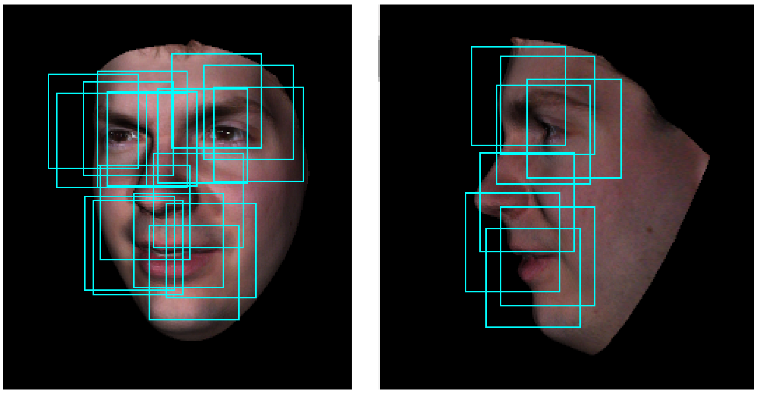

Figure 4.5: Example showing regions around facial landmark points for frontal poses (left image) and side poses (right image) from which DCT features are extracted. For visibility reasons only a part of the used points is displayed.

To identify features achieving the highest recognition accuracy, extensive tests are performed (see section 5).

Since it is assumed that information captured by shape features is different from information captured by appearance features, it is proposed to combine these two types of features for further improvement of recognition accuracy.

Therefore, shape feature vectors and appearance feature vectors are concatenated. Combinations tested are: shape coordinates + SIFT descriptors, and shape coordinates + DCT features. Shape and appearance parameters were not considered for combinations, since recognition accuracy was much lower compared to shape coordinates.

Combined shape coordinates + SIFT descriptors produce a 8704 + 166 = 8870-dimensional vector for frontal poses and a 4608 + 166 = 4774-dimensional vector for side poses, while combined shape coordinates + DCT feature vectors have 5440 + 166 = 5606 dimensions for frontal poses and 2880 + 166 = 3046 dimensions for side poses.

In order to reduce dimensionality of these high-dimensional feature vectors and to improve learnability, different feature selection methods were tested, namely PCA [52], LDA [53], LPP [54] and F-score, but only F-score (explained in section 3.3) showed an improvement in recognition accuracy for most feature types and therefore is used in this work. Feature selection is first performed on the training set to find relevant features. The indices of the selected features are saved, so that the corresponding relevant features in the test set can be extracted easily at runtime.

4.2.3 Facial expression recognition

Features from the vectors, processed by F-score feature selection, are scaled to a range from 0 to 1, in order to achieve better comparability of the features and to avoid features with bigger values dominating others. It is important to apply the same scaling method to both training and test data to produce meaningful results.

The scaled feature vectors serve as input for a pose-dependent linear multi-class SVM (described in section 3.4), which is selected according to pose estimation result (here: groundtruth pose). For this classification step, LibSVM library [55] is used, which offers many options for SVM classification, e.g. linear SVMs or SVMs with different kernels. It also implements the F-score feature selection tool, which is utilized in this work. For classification, linear SVMs are chosen, due to the facts, that using non-linear kernel SVMs requires an extensive grid-search for kernel parameters and that classifying high-dimensional feature vectors with kernel SVMs is much slower than classification with linear SVMs.

Completing the recognition process, the SVM outputs the detected expression class, which can be anger, disgust, fear, happiness, sadness or surprise.

5. Evaluation

For validating the proposed system, extensive experiments were conducted. This section describes the data used for experiments and the experimental setup. Then, results of expression classification using a variety of features are presented.

5.1 Data

In [3], the Binghamton University 3D Facial Expression Database (BU-3DFE) is introduced, which is one of the most commonly used databases for multi-view facial expression recognition.

It contains 3D models of 100 persons with texture and 83 annotated landmark points per model. Subjects in the database are of different age, ranging from 18 to 70 years, and a wide variety of ethnicities/races, including white, black, east-asian, middle-east-asian, hispanic-latino and others. 56% of the subjects are female, 44% male. Each subject shows 7 expressions, which are: neutral, anger, disgust, fear, happiness, sadness and surprise. All subjects display all expressions except neutral at four different levels of intensity from low to high. Consequently, for each subject there are 25 3D models present, which results in an overall number of 2500 facial expression models.

For the system presented in this thesis, 2D images of facial expressions, taken from different view angles, are needed. Therefore, 3D models from the database are rendered together with the texture using VTK (The Visualization Toolkit). The models are rotated at yaw angle from -90 to +90 degrees in steps of 15 degrees. For every step, an image together with the coordinates of the landmark points is saved, resulting in 13 images per face model. After repeating this procedure for every model from the database, 13 poses are available, each containing 2500 images and 2500 sets of landmark points, which adds up to 32500 data elements (image + landmarks). The extracted images have a resolution of 300 × 300 pixels.

Examples of the database are shown in Fig. 5.1, 5.2 and 5.3.

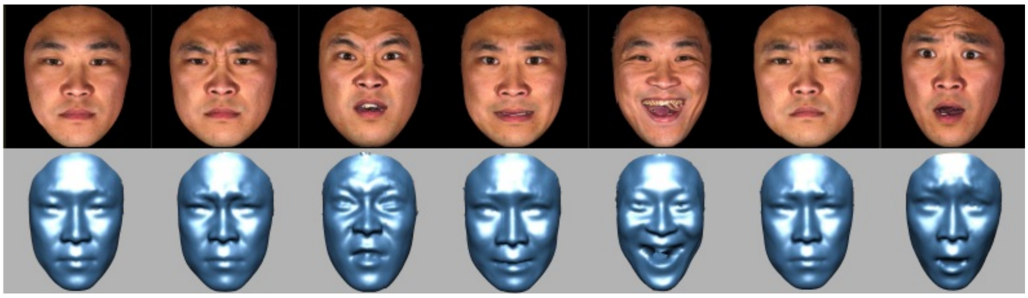

Figure 5.1: Example of BU-3DFE data. Shown expressions from left to right are: neutral, anger, disgust, fear, happiness, sadness, surprise.

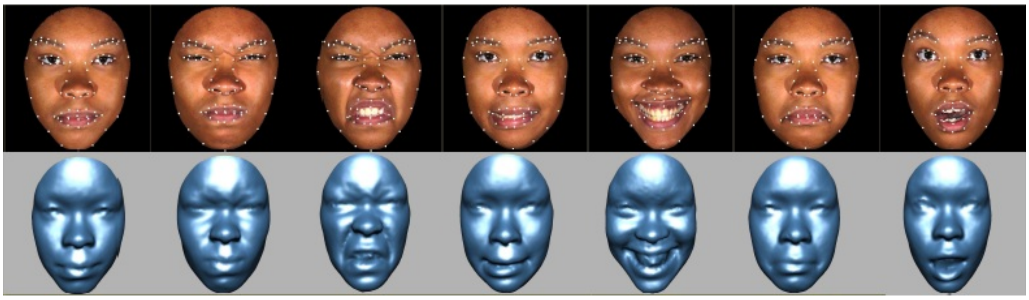

Figure 5.2: Example of BU-3DFE data including landmark points.

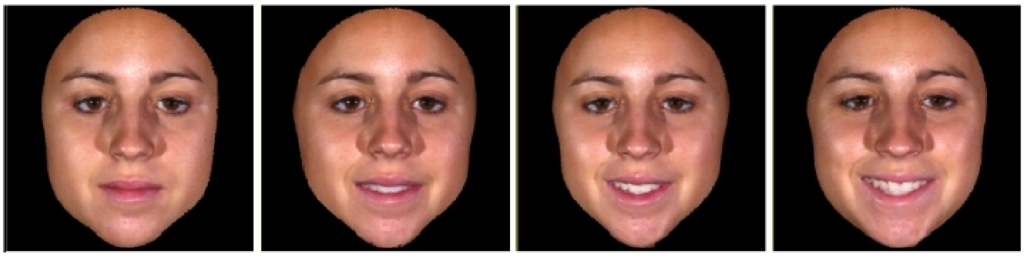

Figure 5.3: Example subject from the BU-3DFE database showing different levels of intensity for expression class happiness.

5.2 Experiment Setup

For every pose, following steps are performed:

- Data taken from the BU-3DFE database is divided into three sets of similar size, two sets containing 33 subjects, one set containing 34 subjects. One set is used for AAM training, one for SVM training and the last one for testing. An AAM is trained using the according set, which contains 2D images and 2D coordinates of facial landmark points.

- Then, for every image of the current SVM training set, AAM fitting is performed using the previously trained AAM. For initialization of the AAM fitting, groundtruth information is used. After extracting shape and appearance features for all images of the set, feature selection is performed and information on relevant features is saved for later usage on the test set. After being processed by feature selection, the data is scaled, and the scaling range is saved for scaling the test set. Finally, the scaled SVM training set is used for training a SVM to recognize six expression classes.
- Similar to the SVM training set, AAM fitting is done on the test set, initialized at groundtruth location and scale. Features are extracted and the relevant ones, determined by previous feature selection on the SVM training set, are selected from the test set and scaled, utilizing the saved range data. This set is then fed to the SVM, which computes the classification accuracy by comparing the classification results with the provided groundtruth labels. Afterwards, statistical data is extracted, e.g. confusion matrices or recognition accuracies for expressions at different intensity levels.

In order to validate the experiments, all possible combinations of set-arrangements are processed, altogether six runs, and the results are averaged.

5.3 Results

Many results were produced, utilizing AAM landmark coordinates, AAM shape and appearance parameters, SIFT and DCT appearance descriptors, as well as combinations of AAM features with SIFT/DCT features for classification. In this section, average recognition accuracies for different features are presented. Results in the different sections are presented in two ways: first, for each expression, the average accuracy over all poses is presented and secondly, for each pose, the average accuracy over all expressions is shown, together with the overall average accuracy. Additionally, the influence of intensity levels on recognition accuracy has been examined and is also presented.

5.3.1 AAM features

From an AAM, which is fitted to an unknown face, information about that face can be extracted, e.g. 2D-coordinates of landmark points, shape parameters and appearance parameters. Landmark coordinates are normalized and will be called shape coordinates from now on. For each face, there are 83 landmark points, which means that the vector containing shape coordinates has 166 dimensions.

In this section, recognition accuracies of classification using shape coordinates (*sc*), shape parameters (*sp*) and appearance parameters (*ap*) as features are compared.

Expression	Shape coordinates	Shape parameters	Appearance parameters
Anger	**70,1**	47,1	62,4
Disgust	**68,4**	49,6	57,8
Fear	**57,7**	38,1	46,6
Happiness	**72,8**	58,7	63,8
Sadness	**67,7**	49,4	58,7
Surprise	**80,8**	67,2	72,8
Overall	**69,6**	51,7	60,4

Table 5.1: Recognition accuracies for different expressions, averaged over all poses for different types of features extracted from AAM.

Pose	Shape coordinates	Shape parameters	Appearance parameters
90l	**69,6**	43,1	54,0
75l	**70,7**	49,2	58,8
60l	**71,3**	50,4	61,6
45l	**71,0**	55,0	62,4
30l	**69,9**	55,0	62,3
15l	**68,7**	56,0	63,9
frontal	**67,0**	54,4	61,5
15r	**67,9**	48,4	61,0
30r	**69,3**	56,8	63,6
45r	**69,3**	55,0	61,7
60r	**69,6**	54,0	61,1
75r	**69,6**	47,4	59,2
90r	**70,6**	47,6	53,9
Overall	**69,6**	51,7	60,4

Table 5.2: Recognition accuracy for different poses, averaged over all emotions for different types of features extracted from AAM; l = left, r = right, numbers represent the degree of view rotation.

In Table 5.1 average recognition rates are presented for each expression. Surprise expression achieves best results for all features, with 80,8% accuracy (*sc*), 67,2% (*sp*) and 72,8% (*ap*). Worst results are obtained for expression fear, with 57,7%(*sc*), 38,1% (*sp*) and 46,6% (*ap*). The other expression classes achieve recognition rates around 70% (*sc*), around 50% (*sp*) and around 60% (*ap*). Detailed results for different poses are displayed in Table 5.2 and Fig. 5.4. For shape coordinates, the accuracy

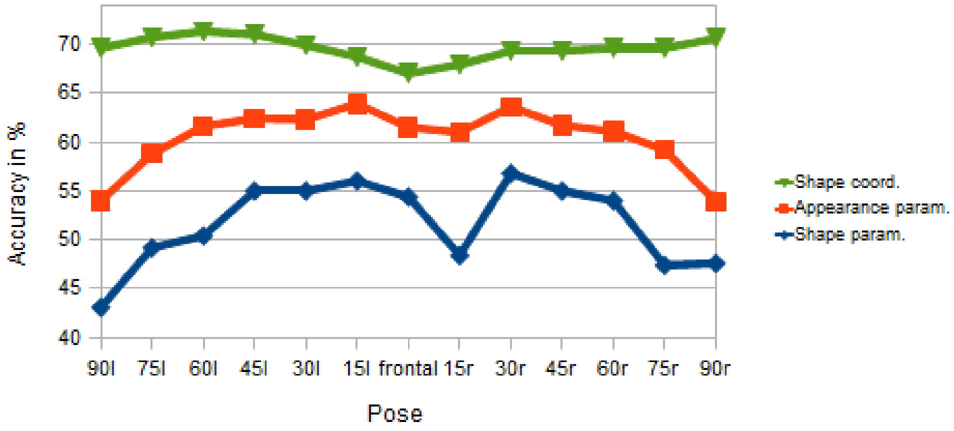

Figure 5.4: Illustration of recognition accuracies across different poses for features extracted from the fitted AAM.

is quite consistent through all poses, it even slightly increases for side views, whereas shape and appearance parameters show a considerable decrease of accuracy for poses near profile view. Best results are obtained between 45l to 75l for shape coordinates while for shape and appearance parameters highest rates are achieved at 15l and 30r, where 'l' and 'r' stand for left, respectively right and the number in front indicates the head rotation in degrees. Comparing classification results for different features, it becomes apparent that *sc* performs best with an overall accuracy of nearly 70%, followed by *ap* with around 60% and finally, *sp* with slightly above 50%.

5.3.2 SIFT/DCT features

Recent approaches to expression recognition use appearance features for classification [5, 6, 41, 42], like e.g. SIFT, HoG [56] and LBP [57]. In this work, the use of SIFT and DCT is explored, also in conjunction with F-score feature selection.

For DCT, a region around landmarks of size 64 × 64 pixels, divided into 4 blocks of 32 × 32 pixels is chosen, and for each block the 20 highest rated coefficients are extracted.

Expression	SIFT	FSEL (SIFT)	DCT	FSEL (DCT)
Anger	67,6	67,2	70,9	**71,3**
Disgust	68,6	69,4	67,9	**70,7**
Fear	53,7	55,1	54,5	**55,7**
Happiness	74,5	77,0	75,0	**78,5**
Sadness	66,4	69,5	67,6	**71,7**
Surprise	83,8	**86,4**	84,0	**86,4**
Overall	69,1	70,8	70,0	**72,4**

Table 5.3: Recognition accuracies for different expressions, averaged over all poses for different types of appearance features, computed at automatically located landmark points. FSEL means that feature selection was applied.

Pose	SIFT	FSEL (SIFT)	DCT	FSEL (DCT)
90l	67,8	**70,1**	65,7	68,1
75l	66,3	68,2	70,6	**72,6**
60l	68,8	70,3	70,7	**72,7**
45l	69,0	70,5	71,1	**73,4**
30l	71,9	**73,2**	71,3	**73,2**
15l	71,0	72,3	71,4	**73,5**
frontal	69,5	70,5	71,9	**73,6**
15r	70,3	72,5	70,3	**72,6**
30r	71,5	73,5	73,2	**75,5**
45r	68,2	70,0	70,9	**73,5**
60r	70,0	70,9	69,7	**73,0**
75r	67,7	69,9	68,0	**71,2**
90r	66,6	**68,2**	65,0	68,0
Overall	69,1	70,8	70,0	**72,4**

Table 5.4: Recognition accuracy for different poses, averaged over all expressions for different types of appearance features, computed at automatically located landmark points. FSEL = feature selection.

In Table 5.3, the appearance features extracted from fitted AAM landmarks are used for classification, and results are shown for original sets of appearance descriptors as well as for sets processed by F-score feature selection method. We can see that there is a constant increase of accuracy if feature selection is applied by up to 4%. Again, surprise and fear show best, respectively worst results for all features, with

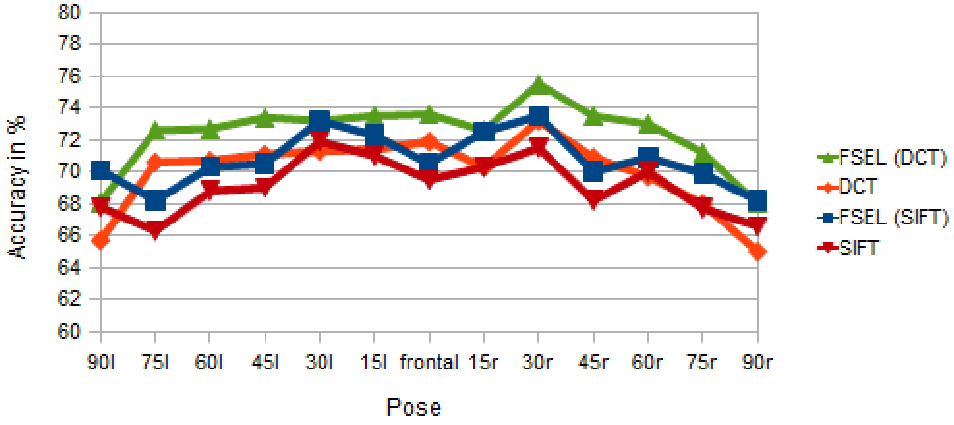

Figure 5.5: Illustration of recognition accuracies across different poses for appearance features extracted at automatically located landmark points. FSEL = feature selection.

around 85% for surprise and around 55% for fear. Results for appearance features for different poses are displayed in Table 5.4 and Fig. 5.5, showing a small advantage of DCT over SIFT. While DCT accuracy drops for profile views and is higher for frontal views, SIFT seems to be more stable over different poses, yet less accurate overall. Highest accuracies are shown at poses 30l and 30r for SIFT, at frontal and 30r for DCT and at 30r for DCT with feature selection. Although DCT has been introduced many years ago, it shows superior results to SIFT in these experiments, with an overall average of 70% (original DCT) and 72,4% (DCT with feature selection), whereas 69,1% (original) and 70,8% (with feature selection) are achieved for SIFT features. Remarkably, the results for shape coordinates from the previous section are slightly better than those of SIFT features at expressions anger, fear and sadness and better than DCT at anger.

Influence of AAM fitting errors on recognition

SIFT and DCT features are also extracted from regions around *groundtruth* landmark points in order to find out how big an influence of possible AAM fitting errors on the classification results is. Recognition rates are shown in Table 5.5.

Expression	GND SIFT	FSEL (GND SIFT)	GND DCT	FSEL (GND DCT)
Anger	71,0	70,7	75,6	**76,0**
Disgust	68,5	69,6	70,3	**72,1**
Fear	55,2	56,2	57,2	**58,6**
Happiness	76,8	78,3	79,1	**80,6**
Sadness	70,8	73,5	74,1	**76,2**
Surprise	85,3	86,7	86,2	**87,0**
Overall	71,3	72,5	73,7	**75,1**

Table 5.5: Recognition accuracies for different expressions, averaged over all poses for different types of appearance features, computed at groundtruth landmark points (GND).

Pose	GND SIFT	FSEL (GND SIFT)	GND DCT	FSEL (GND DCT)
90l	71,5	71,7	71,3	**72,2**
75l	71,3	72,4	71,9	**74,1**
60l	69,1	70,2	73,5	**74,5**
45l	70,2	71,5	73,7	**75,2**
30l	73,4	74,0	76,4	**76,7**
15l	71,7	73,2	76,2	**77,8**
frontal	71,9	72,5	75,4	**77,2**
15r	73,0	73,8	74,3	**76,5**
30r	73,5	74,8	75,8	**77,7**
45r	70,2	71,0	73,4	**75,4**
60r	71,1	72,7	73,2	**74,0**
75r	69,9	72,3	72,0	**72,9**
90r	69,5	**72,4**	71,2	71,5
Overall	71,3	72,5	73,7	**75,1**

Table 5.6: Recognition accuracies for different poses, averaged over all expressions for different types of appearance features, computed at groundtruth landmark points (GND).

Compared to appearance features from AAM landmarks, accuracy increases especially for expressions sadness (up to 6,5%) and anger (up to 5%), while improvements on disgust, fear and surprise are smaller. Table 5.6 and Fig. 5.6 display classification results for different poses, showing that an increase of head rotation leads to a decrease of accuracy for DCT, similar to features extracted at automatically selected landmark points. SIFT features show more stable results for profile poses. For both DCT and SIFT, best results are shown at frontal views from 30l to 30r. Overall, DCT features with feature selection show best results so far, with an average recognition rate of 75,1%, followed by original DCT features (73,7%), SIFT features with feature selection(72,5%) and original SIFT features (71,3%). Thus, the overall average results are decreased by up to 3,7% by misplaced landmarks through AAM fitting errors.

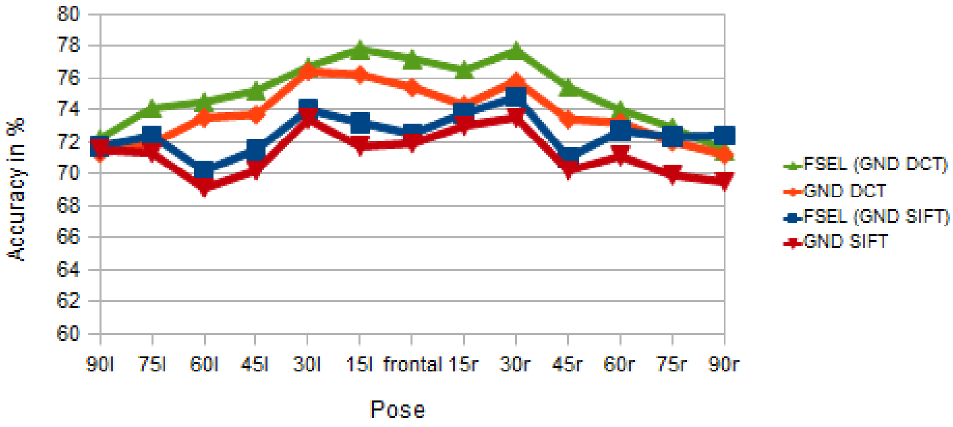

Figure 5.6: Illustration of recognition accuracies across different poses for appearance features extracted at groundtruth landmark points. FSEL = feature selection.

5.3.3 Combination of features

It is further investigated whether or not recognition results can be improved by combining different types of features. Therefore, shape coordinates obtained from fitted AAMs are concatenated with appearance descriptors, which were computed on these coordinates, to form a new feature vector. Results are also displayed for feature vectors on which feature selection was performed. Compared to SIFT features only, recognition rates for expressions increase constantly by around 1% when using a combination of shape coordinates and SIFT features (see Table 5.7). An improvement of up to 2,5% is observed for DCT features (original and with feature selection). Classification results for combined features at different poses in Table 5.8 show the same characteristics as results for appearance features only. Compared to that, overall average recognition rates improve by up to 2% to 74,1% at maximum for shape coordinates + DCT with feature selection.

Expression	Shape coordinates + SIFT	FSEL (Shape coordinates + SIFT)	Shape coordinates + DCT	FSEL (Shape coordinates + DCT)
Anger	69,2	69,3	73,4	**74,0**
Disgust	69,5	70,7	70,3	**72,3**
Fear	55,3	56,7	56,1	**57,8**
Happiness	75,8	78,2	77,1	**80,3**
Sadness	67,9	70,7	69,3	**73,5**
Surprise	84,5	86,4	85,0	**86,8**
Overall	70,4	72,0	71,9	**74,1**

Table 5.7: Recognition accuracies for different expressions, averaged over all poses for combinations of shape coordinates with appearance features (SIFT/DCT), FSEL = feature selection.

Pose	Shape coordinates + SIFT	FSEL (Shape coordinates + SIFT)	Shape coordinates + DCT	FSEL (Shape coordinates + DCT)
90l	69,6	**71,3**	68,9	**71,3**
75l	68,6	70,5	72,0	**74,2**
60l	70,1	71,5	72,5	**74,1**
45l	70,0	71,8	73,6	**74,8**
30l	72,5	74,2	72,8	**74,8**
15l	72,3	71,4	72,4	**75,0**
frontal	70,1	71,9	73,0	**74,9**
15r	71,4	73,3	72,7	**74,9**
30r	72,2	73,9	73,9	**76,0**
45r	69,2	71,0	73,0	**75,2**
60r	70,8	71,9	71,8	**74,2**
75r	69,2	71,5	70,2	**72,8**
90r	68,7	70,3	67,8	**71,4**
Overall	70,4	72,0	71,9	**74,1**

Table 5.8: Recognition accuracies for different poses, averaged over all expressions for combinations of shape coordinates with appearance features (SIFT/DCT), FSEL = feature selection.

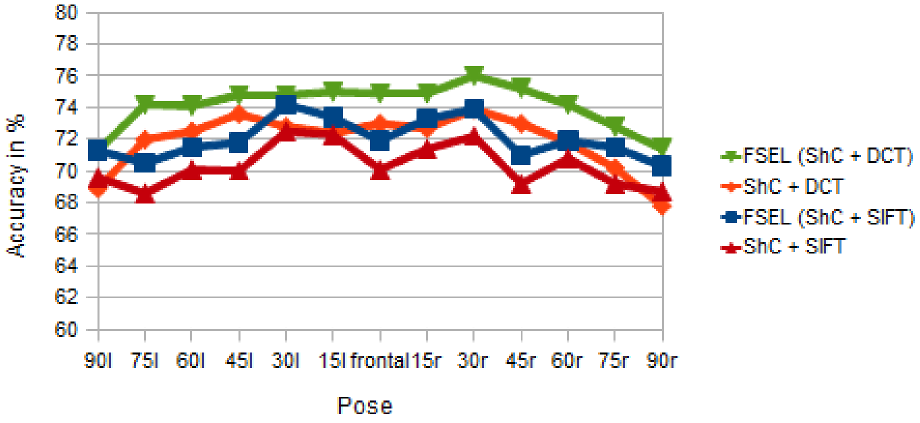

Figure 5.7: Illustration of recognition accuracies across different poses for combinations of shape coordinates (ShC) with appearance features (SIFT/DCT), FSEL = feature selection.

In Figure 5.7 recognition rates across different poses are illustrated. We can see that the improvement through the application of feature selection is quite constant. Displayed in Table 5.9 is a complete confusion matrix for expression classification using a combination of shape coordinates and DCT features with feature selection. Groundtruth expressions are shown on the left and recognized expressions on the top. Most easily confused are following pairs of expressions: anger and sadness (13,5% / 17,1%), anger and disgust (10,4% / 6,6%), fear and disgust (9,9% / 6,8%) and fear and happiness (14,5% / 12,3%).

	Anger	Disgust	Fear	Happiness	Sadness	Surprise
Anger	**74.0**	6.6	3.9	1.4	13.5	0.6
Disgust	10.4	**72.3**	6.8	4.6	2.7	3.3
Fear	6.3	9.9	**57.8**	14.5	5.4	6.1
Happiness	1.8	4.3	12.3	**80.3**	0.6	0.7
Sadness	17.1	2.6	5.4	1.0	**73.5**	0.5
Surprise	1.7	3.1	6.0	1.3	1.0	**86.8**

Table 5.9: Complete confusion matrix for combination of shape coordinates with DCT. Feature selection was applied.

5.3.4 Influence of intensity on recognition

In this section, recognition rates are displayed for each of the four intensity levels for combinations of shape and appearance features. Experimental setup and data is the same as above. The shown percentages are average accuracies over all expressions and poses.

	Shape coordinates + SIFT	FSelect (Shape coordinates + SIFT)	Shape coordinates + DCT	FSelect (Shape coordinates + DCT)
Intensity level 1	61,0	62,8	62,2	**64,2**
Intensity level 2	70,7	72,3	72,1	**74,7**
Intensity level 3	73,8	75,2	75,7	**77,8**
Intensity level 4	76,0	77,8	77,6	**79,7**

Table 5.10: Recognition rates for classification of data containing expressions at different levels of intensity. Averaged over all expressions and poses.

As expected, an increase in the intensity level leads to an increase of accuracy as can be seen in Table 5.10. Already from lowest to second lowest level, a big improvement is shown (about 10%), while from second lowest to highest level, the increase of accuracy is not as significant (about 5% from level 2 to level 4).

Figure 5.8 shows recognition rates across different poses for different intensity levels, indicating that differences between accuracies of different intensity levels remain quite constant across changing poses.

In initial experiments, intensity-specific classifiers were trained, but no improvement of accuracy was gained.

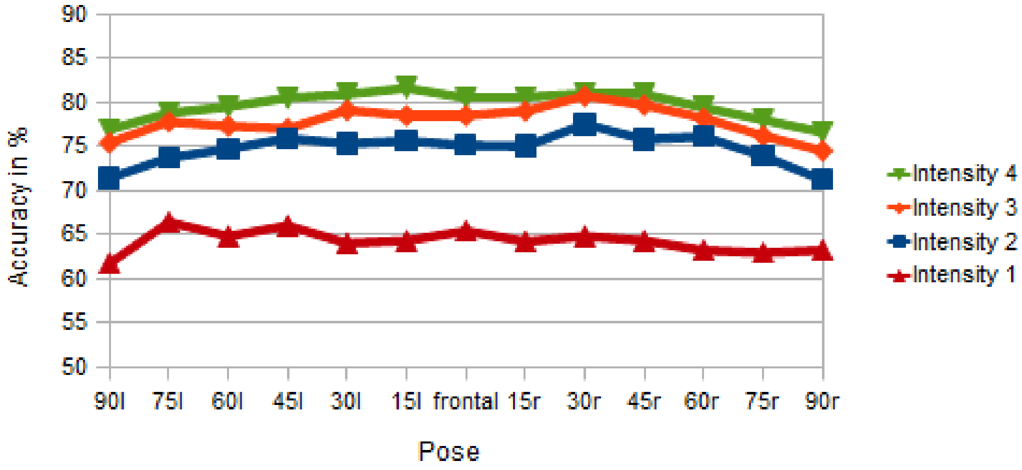

Figure 5.8: Illustration of accuracies for different poses for combination of shape coordinates and DCT features with feature selection at different intensity levels.

6. Conclusion

In this thesis, a multi-view facial expression recognition system was developed with the capability of being applied to real-world situations. Therefore, facial landmarks are found automatically on faces showing pose variations up to profile view. In order to handle these tasks, pose-dependent Active Appearance Models and pose-dependent Support Vector Machines are trained. Pose-dependent means, that for each pose, one AAM/SVM is trained. The appropriate AAM/SVM for a new input image is determined by a pose estimator. For a given input image, a pose-dependent AAM is initialized at a location obtained from a face detector. In this work, groundtruth information was used in order not to influence the recognition results by face detection or pose estimation. The AAM is then fitted to the face, providing locations of facial landmarks. On these landmarks, local appearance features (SIFT and DCT) are computed. The extracted features form a feature vector, on which feature selection is performed. Then, this vector is passed to a previously trained pose-dependent SVM, which outputs the recognized expression class.

For extensive experimental evaluation, the BU-3DFE database was used to construct sets of 2D images from 100 subjects, containing six expression classes at 13 different poses. Classification results for different features and combinations of features were analyzed, and the use of feature selection methods was explored. It was shown, that for features extracted from the fitted AAM, normalized landmark coordinates (69,6%) achieve much higher recognition rates than shape parameters (51,7%) and appearance parameters (60,4%). A comparison of SIFT and DCT appearance features indicated higher accuracy for DCT features (reduced DCT: 72,4%; reduced SIFT: 70,8%). The effect of AAM fitting errors, leading to misplacements of facial landmarks, on recognition accuracy was investigated. Therefore, results for appearance features extracted at automatically located landmarks were compared to features extracted at groundtruth landmarks, with the latter showing an improvement of up to 2,2% for SIFT and up to 3,7% for DCT features. By combining shape and appearance features, recognition accuracy increased in comparison to the use of single feature types. Best results for features extracted at automatically located landmark points were achieved by a combination of shape coordinates and DCT features with feature selection at an overall average recognition rate of 74,1%.

The influence of the displayed expression intensity level on recognition accuracy was also studied, showing a recognition rate on lowest intensity level (around 65% for shape coordinates + DCT) much lower than on highest intensity level (around 80%). The improvement from lowest to second lowest intensity was already quite big, with an increase of accuracy by about 10%.

There are several issues for future work. The addition of a state-of-the-art face-detector and a pose-estimator to the system is necessary to utilize the system in real-world applications. Additional landmark points, e.g. at nasolabial furrows, possibly carry information, which is relevant for expression recognition, and should be considered to be utilized. Also, the use of different classifiers, as well as varying the size of training and test sets could be interesting.

Bibliography

[1] P. Ekman and W. V. Friesen, “Constants across cultures in the face and emotion,” *Journal of Personality and Social Psychology*, vol. 17, no. 2, pp. 124 – 129, 1971.

[2] P. Ekman and W. V. Friesen, *Facial Action Coding System.* Consulting Psychologists Press, 1978.

[3] L. Yin, X. Wei, Y. Sun, J. Wang, and M. Rosato, “A 3D facial expression database for facial behavior research,” in *7th International Conference on Automatic Face and Gesture Recognition*, pp. 211 – 216, 2006.

[4] Y. Hu, Z. Zeng, L. Yin, X. Wei, J. Tu, and T. Huang, “A study of non-frontal-view facial expressions recognition,” in *19th International Conference on Pattern Recognition*, 2008.

[5] Y. Hu, Z. Zeng, L. Yin, X. Wei, X. Zhou, and T. Huang, “Multi-view facial expression recognition,” in *8th IEEE International Conference on Automatic Face and Gesture Recognition*, 2008.

[6] W. Zheng, H. Tang, Z. Lin, and T. Huang, “A novel approach to expression recognition from non-frontal face images,” in *IEEE 12th International Conference on Computer Vision*, pp. 1901 – 1908, 2009.

[7] O. Rudovic, I. Patras, and M. Pantic, “Regression-based multi-view facial expression recognition,” in *International Conference on Pattern Recognition*, 2010.

[8] S. Taheri, P. Turaga, and R. Chellappa, “Towards view-invariant expression analysis using analytic shape manifolds,” in *Automatic Face and Gesture Recognition*, 2011.

[9] M. Pantic and I. Patras, “Dynamics of facial expression: recognition of facial actions and their temporal segments from face profile image sequences,” *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 36, no. 2, pp. 433 – 449, 2006.

[10] J. Sung and D. Kim, “Pose-robust facial expression recognition using view-based 2D + 3D AAM,” *IEEE Transactions on Systems, Man and Cybernetics - Part A: Systems and Humans*, vol. 38, pp. 852 – 866, 2008.

[11] T. F. Cootes, G. J. Edwards, and C. J. Taylor, "Active Appearance Models," in *Proceedings of the 5th European Conference on Computer Vision-Volume II*, pp. 484 – 498, Springer-Verlag, 1998.

[12] D. G. Lowe, "Object recognition from local scale-invariant features," in *IEEE International Conference on Computer Vision*, vol. 2, (Los Alamitos, CA, USA), p. 1150, IEEE Computer Society, 1999.

[13] N. Ahmed, T. Natarajan, and K. R. Rao, "Discrete Cosine Transfom," *IEEE Transactions on Computers*, vol. 23, pp. 90 – 93, 1974.

[14] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, pp. 273 – 297, 1995.

[15] H. Tang and T. S. Huang, "3D facial expression recognition based on properties of line segments connecting facial feature points," in *8th IEEE International Conference on Automatic Face and Gesture Recognition*, 2008.

[16] H. Tang and T. Huang, "3D facial expression recognition based on automatically selected features," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 2008.

[17] H. Soyel and H. Demirel, "Facial expression recognition using 3D facial feature distances," in *Image Analysis and Recognition*, vol. 4633 of *Lecture Notes in Computer Science*, pp. 831 – 838, Springer Berlin / Heidelberg, 2007.

[18] H. Soyel and H. Demirel, "3D facial expression recognition with geometrically localized facial features," in *23rd International Symposium on Computer and Information Sciences*, 2008.

[19] U. Tekguc, H. Soyel, and H. Demirel, "Feature selection for person-independent 3D facial expression recognition using NSGA-II," in *24th International Symposium on Computer and Information Sciences*, pp. 35 – 38, 2009.

[20] H. Soyel and H. Demirel, "Optimal feature selection for 3D facial expression recognition using coarse-to-fine classification," *Turk J Elec Eng & Comp Sci*, vol. 18, No.6, pp. 1031 – 1040, 2010.

[21] B. Fasel and J. Luettin, "Automatic facial expression analysis: a survey," *Pattern Recognition*, vol. 36, no. 1, pp. 259 – 275, 2003.

[22] Y. Zhu, F. De la Torre, J. Cohn, and Y.-J. Zhang, "Dynamic cascades with bidirectional bootstrapping for spontaneous facial action unit detection," in *3rd International Conference on Affective Computing and Intelligent Interaction and Workshops*, 2009.

[23] T. Simon, M. H. Nguyen, F. De la Torre, and J. Cohn, "Action unit detection with segment-based SVMs," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2737 – 2744, 2010.

[24] F. Zhou, F. De la Torre, and J. Cohn, "Unsupervised discovery of facial events," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2574 – 2581, 2010.

[25] M. S. Bartlett, G. Littlewort, I. Fasel, and J. R. Movellan, "Real time face detection and facial expression recognition: Development and applications to human computer interaction.," in *Computer Vision and Pattern Recognition Workshop*, vol. 5, (Los Alamitos, CA, USA), p. 53, IEEE Computer Society, 2003.

[26] Z. Zhang, M. Lyons, M. Schuster, and S. Akamatsu, "Comparison between geometry-based and Gabor-Wavelets-based facial expression recognition using Multi-Layer Perceptron," in *Proceedings of the 3rd International Conference on Face and Gesture Recognition*, 1998.

[27] P. Yang, Q. Liu, and D. Metaxas, "Exploring facial expressions with compositional features," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2638 – 2644, 2010.

[28] B. Jiang, M. Valstar, and M. Pantic, "Action unit detection using sparse appearance descriptors in space-time video volumes," in *Proceedings of IEEE International Conference on Automatic Face and Gesture Recognition*, 2011.

[29] C. Shan and T. Gritti, "Learning discriminative LBP-histogram bins for facial expression recognition," in *Proceedings of British Machine Vision Conference*, 2008.

[30] N. Sebe, M. Lew, Y. Sun, I. Cohen, T. Gevers, and T. Huang, "Authentic facial expression analysis," *Image and Vision Computing*, vol. 25, no. 12, pp. 1856 – 1863, 2007.

[31] S. Ramanathan, A. Kassim, Y. V. Venkatesh, and W. S. Wah, "Human facial expression recognition using a 3D Morphable Model," in *IEEE International Conference on Image Processing*, pp. 661 – 664, 2006.

[32] S. Lucey, I. Matthews, C. Hu, Z. Ambadar, F. De la Torre, and J. Cohn, "AAM derived face representations for robust facial action recognition," in *Proceedings of the 7th International Conference on Automatic Face and Gesture Recognition*, pp. 155 – 162, 2006.

[33] P. Lucey, J. Cohn, T. Kanade, J. Saragih, Z. Ambadar, and I. Matthews, "The extended Cohn-Kanade dataset (CK+): A complete dataset for action unit and emotion-specified expression," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pp. 94 – 101, 2010.

[34] C. Hu, Y. Chang, R. Feris, and M. Turk, "Manifold based analysis of facial expression," in *Computer Vision and Pattern Recognition Workshop*, vol. 5, (Los Alamitos, CA, USA), p. 81, IEEE Computer Society, 2004.

[35] Y. Tong, J. Chen, and Q. Ji, "A unified probabilistic framework for spontaneous facial action modeling and understanding," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 2, pp. 258 – 273, 2010.

[36] I. Mpiperis, S. Malassiotis, and M. Strintzis, "Bilinear models for 3D face and facial expression recognition," *IEEE Transactions on Information Forensics and Security*, vol. 3, no. 3, pp. 498 – 511, 2008.

[37] C.-S. Lee and A. Elgammal, "Nonlinear shape and appearance models for facial expression analysis and synthesis," in *Proceedings of the 18th International Conference on Pattern Recognition*, vol. 1, pp. 497 – 502, 2006.

[38] Y. Chang, M. Vieira, M. Turk, and L. Velho, "Automatic 3D facial expression analysis in videos," in *Analysis and Modelling of Faces and Gestures*, vol. 3723 of *Lecture Notes in Computer Science*, pp. 293 – 307, Springer Berlin / Heidelberg, 2005.

[39] X. Zhao, D. Huang, E. Dellandrea, and L. Chen, "Automatic 3D facial expression recognition based on a Bayesian Belief Net and a Statistical Facial Feature Model," in *International Conference on Pattern Recognition*, pp. 3724 – 3727, 2010.

[40] W.-K. Liao and G. Medioni, "3D face tracking and expression inference from a 2D sequence using manifold learning," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[41] W. Zheng, H. Tang, Z. Lin, and T. Huang, "Emotion recognition from arbitrary view facial images," in *Computer Vision - ECCV 2010*, vol. 6316 of *Lecture Notes in Computer Science*, pp. 490 – 503, Springer Berlin / Heidelberg, 2010.

[42] S. Moore and R. Bowden, "Local Binary Patterns for multi-view facial expression recognition," *Computer Vision and Image Understanding*, vol. 115, no. 4, pp. 541 – 558, 2011.

[43] I. Matthews and S. Baker, "Active Appearance Models revisited," *International Journal of Computer Vision*, vol. 60, pp. 135 – 164, 2004.

[44] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, pp. 91 – 110, 2004.

[45] G. Strang, "The Discrete Cosine Transform," *SIAM Review*, vol. 41, pp. 135 – 147, 1999.

[46] H. K. Ekenel and R. Stiefelhagen, "Local appearance based face recognition using Discrete Cosine Transform," in *13th European Signal Processing Conference Antalya, Turkey*, 2005.

[47] H. Gao, "Face registration with Active Appearance Models for local appearance-based face recognition," Diplomarbeit, Interactive Systems Labs, Universität Karlsruhe (TH), June 2008.

[48] Y.-W. Chen and C.-J. Lin, "Combining SVMs with various feature selection strategies," in *Feature Extraction*, vol. 207 of *Studies in Fuzziness and Soft Computing*, pp. 315 – 324, Springer Berlin / Heidelberg, 2006.

[49] http://de.wikipedia.org/wiki/Support_Vector_Machine, March 2011.

[50] http://courses.media.mit.edu/2006fall/mas622j/Projects/aisen-project/index.html, March 2011.

[51] S. Z. Li, A. K. Jain, Y.-L. Tian, T. Kanade, and J. F. Cohn, "Facial expression analysis," in *Handbook of Face Recognition*, pp. 247 – 275, Springer New York, 2005.

[52] K. Pearson, "On lines and planes of closest fit to systems of points in space," *Philosophical Magazine Series 6*, vol. 2, no. 11, pp. 559 – 572, 1901.

[53] K. Fukunaga, *Introduction to statistical pattern recognition (2nd ed.)*. San Diego, CA, USA: Academic Press Professional, Inc., 1990.

[54] X. He and P. Niyogi, "Locality Preserving Projections," *Advances in Neural Information Processing Systems*, vol. 16, pp. 153 – 160, 2003.

[55] C.-C. Chang and C.-J. Lin, *LIBSVM: a library for support vector machines*, 2001. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[56] N. Dalal and B. Triggs, "Histograms of Oriented Gradients for human detection," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 886 – 893, 2005.

[57] T. Ojala, M. Pietikäeinen, and D. Harwood, "A comparative study of texture measures with classification based on featured distributions," *Pattern Recognition*, vol. 29, no. 1, pp. 51 – 59, 1996.